



HAL
open science

Estimation des paramètres de transport en milieu hydro-géologique et analyse d'incertitudes

Thomas Migliore

► **To cite this version:**

Thomas Migliore. Estimation des paramètres de transport en milieu hydro-géologique et analyse d'incertitudes. Mathématiques [math]. Université Nice Sophia Antipolis, 2009. Français. NNT : . tel-00442635v3

HAL Id: tel-00442635

<https://theses.hal.science/tel-00442635v3>

Submitted on 28 Dec 2009

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

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE NICE SOPHIA ANTIPOLIS – UFR Sciences

École Doctorale Sciences Fondamentales et Appliquées

THÈSE

pour obtenir le titre de

Docteur en SCIENCES

Spécialité :

MATHÉMATIQUES

présentée et soutenue par

Thomas MIGLIORE

Estimation des paramètres de transport dans un milieu hydro-géologique et analyse d'incertitudes.

Thèse dirigée par Jacques BLUM & Didier AUROUX

soutenue le 7 décembre 2009

devant le jury composé de :

M. Didier AUROUX	Université de Nice	Directeur de thèse
M. Jacques BLUM	Université de Nice	Directeur de thèse
M. Laurent HASCOËT	INRIA Sophia Antipolis	Examineur
M. Jérôme JAFFRÉ	INRIA Rocquencourt	Rapporteur
M. Mohamed JAOUA	Université de Nice	Examineur
M. Michel KERN	INRIA Rocquencourt	Examineur
M. Laurent LOTH	ANDRA	Examineur
M. Bernard SARAMITO	Université de Clermont-Ferrand	Rapporteur

Laboratoire Jean-Alexandre Dieudonné, Parc Valrose, 06108 NICE Cedex 2

« La raison fait l'homme mais c'est le sentiment qui le conduit. »

Jean-Jacques Rousseau

A mon grand-père,

Remerciements

Tout d'abord, je tiens à remercier mes directeurs de thèse, Jacques Blum, sans qui je n'aurais pas commencé cette thèse, et Didier Auroux pour son aide précieuse en informatique. J'ai eu la chance de pouvoir travailler dans d'excellentes conditions matérielles et d'avoir eu, grâce à Jacques et Didier, une grande liberté dans mon travail. Je tiens aussi à remercier toutes les personnes de l'Andra que j'ai sollicitées tout au long de ces trois années : tant le personnel "scientifique" (en particulier Daniel Coehlo, Jérôme Gaombalet et Laurent Loth) qu'administratif (Catherine Bergdoll, Carole Laemmel, Valérie Makangu, Karine Robert). Un grand merci à Laurent Hascoët pour sa gentillesse et sa disponibilité, il a en effet passé beaucoup de temps sur mon code et au développement de Tapenade.

Je remercie ensuite l'ensemble des membres du jury, qui m'ont fait l'honneur de bien vouloir examiner mon travail : Jérôme Jaffré et Bernard Saramito pour avoir accepté d'être rapporteurs et avoir pris le soin de lire attentivement mon manuscrit ; Laurent Hascoët et Laurent Loth pour avoir accepté d'examiner cette thèse et enfin Michel Kern et Mohamed Jaoua pour avoir accepté de faire partie du jury.

Je tiens à exprimer toute ma reconnaissance au personnel administratif du laboratoire JAD pour leur gentillesse et leur disponibilité, à savoir (par ordre alphabétique) Cécile, Christiane, Claudine, Fernande, Isabelle DA, Isabelle L., Janine, Jean-Louis (le joueur de badminton), Jean-Marc et Julien pour l'informatique (sans qui je n'aurai peut être pas fini ma thèse...), Jean-Paul (Mr Xerox), Mr Maisonobe (notre cher directeur), Stéphanie (et son efficacité pour les missions).

Je remercie également les personnes de l'équipe "EDP et Analyse Numérique" dont certaines se sont montrées très disponibles lorsque je les ai sollicitées. Une mention spéciale pour Mr Florent Berthelin qui me supporte depuis la maîtrise où il a eu l'immense honneur (...) de m'enseigner les joies de l'analyse et avec qui j'ai eu de longues discussions en longeant le couloir circulaire de notre fabuleux 4^{ème} étage. En parlant du 4^{ème}, comme dirait Didier, je tiens à remercier tous ceux (anciens et actuels) qui ont contribué à sa renommée internationale !

Je continue sur ma lancée en remerciant les doctorants du laboratoire avec qui j'ai passé de très bons moments durant ces trois années (certains d'ailleurs mémorables) : Mimi (le beau gosse), Lord of Sol (je sais toujours pas ce que ça veut dire... mais fan inconditionnel de JM!!), Mr Ba (mon inestimable co-bureau), You (ma voisine austro-hongroise), Joan (et ses t-shirts troués) et sans oublier Thomas (et ses rendez-vous chez le kiné), Pat (pour ce moment mémorable à Dourdan), Mr et Mme Olivetti (pour les conseils L^AT_EX et pour m'avoir supporté...), Pierre (et la classe américaine), Cédric (l'homme de la fusion), Damien, Hugues, Laura, Cristina et tous les autres.

Je terminerai en remerciant tout particulièrement mes parents, mes grands-parents, mon frère, mon parrain et Caroline (qui partage ma vie) pour "l'assistance technique, option tout fourni" (assurée depuis plus de 28 ans!) et pour le soutien moral qu'ils m'ont apporté dans les moments difficiles et sans qui je n'en serai pas là aujourd'hui, cette thèse est aussi un peu la leur. J'ai également une pensée pour ma fille Camille qui est née il y a peu pour assister à l'achèvement de cette thèse.

Table des matières

1	Introduction	1
1.1	Contexte général	2
1.1.1	Le stockage des déchets nucléaires : enjeux, outils et difficultés	2
1.1.2	Les différents types de déchets	3
1.1.3	Les lois de 1991 et 2006 et le rôle de l'Andra	4
1.1.4	Le stockage en couche géologique profonde	4
1.2	Contexte scientifique	5
1.2.1	Problématiques liées au stockage des déchets à haute et moyenne activité et à vie longue	6
1.2.2	Le laboratoire souterrain et l'expérimentation DIR	6
1.2.3	Problème inverse et mesures	7
1.3	Modèles physiques et mathématiques	7
1.3.1	Les écoulements en milieu poreux saturé	8
1.3.2	Le transport de radionucléides en milieu poreux saturé	9
1.4	Présentation du problème inverse	11
1.4.1	Estimation des paramètres de transport	12
1.4.2	Identification de zones	12
1.4.3	Aspect implémentation	12
1.5	Plan du mémoire	13
2	Modèles mathématiques et numériques	15
2.1	Les équations d'état	16
2.1.1	L'équation d'écoulement	16
2.1.2	L'équation de transport	17
2.2	Discrétisation des équations	17
2.2.1	Introduction	17
2.2.2	Discrétisation de l'équation d'écoulement	18
2.2.3	Discrétisation de l'équation de transport : le code Traces	20
2.3	Plateforme de calcul Alliances	23
2.3.1	Définition & Historique	23
2.3.2	Fonctionnement de la plate-forme	23
2.3.3	Maillages & Post-traitement	24
2.4	Implémentation	25
2.4.1	Le code Traces	25
2.4.2	La plate-forme Alliances	26
2.5	Position du problème	26
2.5.1	Cadres et objectifs de notre travail	27
2.5.2	Présentation des cas-tests	27
2.6	Résolution du problème	30
2.6.1	Cas-test "colis"	30

2.6.2	Cas-test “réel”	31
3	Etude du problème inverse	35
3.1	Introduction	36
3.1.1	Etat de l’art	36
3.1.2	Cadres et objectifs de notre travail	37
3.2	Etude théorique du problème inverse	38
3.2.1	Rappels de contrôle optimal	38
3.2.2	Le modèle linéaire tangent	39
3.2.3	Introduction de l’état adjoint	40
3.2.4	Obtention formelle de l’équation de transport adjointe	43
3.3	Etude numérique du problème inverse	52
3.3.1	Introduction	52
3.3.2	Le code adjoint	52
3.3.3	Rappels sur les méthodes d’optimisation	53
3.3.4	Description de l’algorithme de minimisation	54
3.3.5	Implémentation	55
4	Différentiation automatique	57
4.1	Généralités sur le calcul des dérivées	58
4.1.1	Introduction	58
4.1.2	Les différentes méthodes de calcul des dérivées	58
4.2	Différentiation automatique du code par Tapenade	62
4.2.1	Généralités sur la différenciation automatique	62
4.2.2	Présentation de Tapenade	64
4.3	Du point de vue numérique	69
4.3.1	Validation des codes générés automatiquement	69
4.4	Conclusion	70
4.4.1	Résumé	70
4.4.2	Discussion sur la différenciation automatique	70
5	Estimation de paramètres	71
5.1	Généralités	72
5.2	Résultats numériques	72
5.2.1	Cas-test “colis”	72
5.2.2	Cas-test “réel”	77
5.3	Conclusion des tests	90
6	Identification de zones	91
6.1	Introduction	92
6.2	Etude numérique	93
6.2.1	Premiers tests	93
6.2.2	Cas concentrique	100
6.3	Conclusion des tests	114
7	Conclusion et perspectives	115
	Annexe	116
	A Nomenclature des grandeurs physiques	117
	Bibliographie	118

Liste des tableaux

2.1	Coefficients pour l'Iode 129	28
2.2	Coefficients de diffusion/dispersion, de porosité et de retard	28
3.1	Algorithme de minimisation	55
4.1	Méthode de la variable complexe	61
4.2	Algorithme de résolution numérique du problème inverse	68
4.3	Précision du test du gradient	69
5.1	Valeurs identifiées pour le cas-test “colis” sans bruit	73
5.2	Valeurs identifiées avec un bruit de 5%	75
5.3	Identification des 6 paramètres	77
5.4	Identification des 2 paramètres de la zone Ω_1	78
5.5	Identification de la porosité	78
5.6	Identification de la diffusion	78
5.7	Identification dans les zones 1 & 3	79
5.8	Identification dans les zones 1 & 3 avec un bruit de 3%	81
5.9	Identification dans les zones 1 & 3 avec un bruit de 5%	83
5.10	Identification dans les zones 1 & 3 avec un bruit de 3%	84
5.11	Identification dans les zones 1 & 3 avec un bruit de 5%	84
5.12	Identification dans les zones 1 & 3 avec un bruit de 10%	87
5.13	Identification dans les zones 1 & 3 avec un bruit de 20%	87
6.1	Valeurs initiales des paramètres	94
6.2	Valeurs initiales des paramètres	97
6.3	Identification dans les zones 1, 3, 4, 5, 6 & 7	101
6.4	Identification dans les zones 1, 3, 4, 5, 6 & 7	103
6.5	Identification de la diffusion avec un bruit de 3%	105
6.6	Identification de la porosité avec un bruit de 3%	106
6.7	Identification dans les zones 1, 3, 4, 5, 6 & 7 avec un bruit de 5%	108
6.8	Identification de la diffusion avec un bruit de 10%	110
6.9	Identification de la porosité avec un bruit de 10%	110
6.10	Identification de la diffusion avec un bruit de 20%	112
6.11	Identification de la porosité avec un bruit de 20%	112
A.1	Valeurs des grandeurs physiques	117

Table des figures

1.1	Laboratoire souterrain de Bure	3
1.2	Structure d'un module	5
2.1	Architecture d'Alliances	24
2.2	Architecture de Traces	25
2.3	Structure du fichier traces0.f90	26
2.4	Représentation du milieu géologique	27
2.5	Représentation du domaine Ω (60 m \times 40 m)	29
2.6	Représentation du domaine Ω (0,4 m \times 2 m)	29
2.7	Chambre de diffusion	29
2.8	Domaine Ω maillé avec 2742 éléments	30
2.9	Visualisation de la concentration obtenue avec Salome à l'instant final	31
2.10	Visualisation de la concentration obtenue avec Matlab à l'instant final	32
2.11	Profil de concentration sur la ligne horizontale γ_1 d'équation $y = 1$	33
2.12	Profil de concentration sur la ligne horizontale γ_2 d'équation $y = 0,9$	34
2.13	Profil de concentration sur la ligne verticale γ_3 d'équation $x = 0$	34
3.1	Domaine Ω	44
3.2	Domaine Ω	48
4.1	Différences finies pour $\log x$ au point 1	60
4.2	Stratégie "Store All"	66
4.3	Checkpointing	67
4.4	Architecture du code adjoint de Traces	68
4.5	Structure du fichier traces0_b.f90	68
4.6	Diagramme de résolution du problème inverse	70
5.1	En haut : évolution des 2 paramètres de porosité en échelle semi-log à gauche. Au milieu à gauche : évolution des 2 paramètres de diffusion en échelle semi-log. Au milieu à droite : évolution de la fonction coût en échelle semi-log. En bas à gauche : évolution des 4 composantes du gradient en échelle semi-log.	74
5.2	En haut : évolution des 2 paramètres de porosité en échelle semi-log à gauche. Au milieu : évolution des 2 paramètres de diffusion en échelle semi-log. En bas à gauche : évolution de la fonction coût en échelle semi-log. En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	76
5.3	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	80

5.4	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	82
5.5	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	83
5.6	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	85
5.7	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	86
5.8	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	88
5.9	En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.	89
6.1	Configuration exacte	94
6.2	Configuration modifiée	94
6.3	Diffusion à l'itération 1, Diffusion à l'itération 947, Diffusion réelle	95
6.4	Porosité à l'itération 1, Porosité à l'itération 947, Porosité réelle	96
6.5	Evolution de la fonction coût (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan)	96
6.6	Evolution du gradient	96
6.7	Evolution de la fonction coût	97
6.8	Evolution du gradient	97
6.9	En haut à gauche : Diffusion à l'itération 1. En bas à droite : Diffusion à l'itération 1857. En bas à gauche : Diffusion à l'itération 2501. En bas à droite : Diffusion réelle.	98
6.10	En haut à gauche : Porosité à l'itération 1. En bas à droite : Porosité à l'itération 1857. En bas à gauche : Porosité à l'itération 2501. En bas à droite : Porosité réelle.	99
6.11	Configuration exacte	100
6.12	Configuration modifiée	100
6.13	Diffusion à l'itération 1, Diffusion à l'itération 179, Diffusion réelle	101
6.14	Porosité à l'itération 1, Porosité à l'itération 179, Porosité réelle	102
6.15	Evolution de la fonction coût	102

6.16	Evolution du gradient	102
6.17	Diffusion à l'itération 1, Diffusion à l'itération 270, Diffusion réelle	104
6.18	Porosité à l'itération 1, Porosité à l'itération 270, Porosité réelle	104
6.19	Evolution de la fonction coût	105
6.20	Evolution du gradient	105
6.21	Diffusion à l'itération 1, Diffusion à l'itération 283, Diffusion réelle	106
6.22	Porosité à l'itération 1, Porosité à l'itération 283, Porosité réelle	107
6.23	Evolution de la fonction coût	107
6.24	Evolution du gradient	107
6.25	Diffusion à l'itération 1, Diffusion à l'itération 299, Diffusion réelle	108
6.26	Porosité à l'itération 1, Porosité à l'itération 299, Porosité réelle	109
6.27	Evolution de la fonction coût	109
6.28	Evolution du gradient	109
6.29	Diffusion à l'itération 1, Diffusion à l'itération 186, Diffusion réelle	111
6.30	Porosité à l'itération 1, Porosité à l'itération 186, Porosité réelle	111
6.31	Evolution de la fonction coût	112
6.32	Evolution du gradient	112
6.33	Diffusion à l'itération 1, Diffusion à l'itération 121, Diffusion réelle	113
6.34	Porosité à l'itération 1, Porosité à l'itération 121, Porosité réelle	113
6.35	Evolution de la fonction coût	114
6.36	Evolution du gradient	114

Chapitre 1

Introduction

Sommaire

1.1 Contexte général	2
1.1.1 Le stockage des déchets nucléaires : enjeux, outils et difficultés	2
1.1.2 Les différents types de déchets	3
1.1.3 Les lois de 1991 et 2006 et le rôle de l'Andra	4
1.1.4 Le stockage en couche géologique profonde	4
1.2 Contexte scientifique	5
1.2.1 Problématiques liées au stockage des déchets à haute et moyenne activité et à vie longue	6
1.2.2 Le laboratoire souterrain et l'expérimentation DIR	6
1.2.3 Problème inverse et mesures	7
1.3 Modèles physiques et mathématiques	7
1.3.1 Les écoulements en milieu poreux saturé	8
1.3.2 Le transport de radionucléides en milieu poreux saturé	9
1.4 Présentation du problème inverse	11
1.4.1 Estimation des paramètres de transport	12
1.4.2 Identification de zones	12
1.4.3 Aspect implémentation	12
1.5 Plan du mémoire	13

On s'intéresse dans cette thèse à l'estimation des paramètres de transport de solutés dans un milieu poreux peu perméable, d'un point de vue théorique et appliqué, en lien avec le stockage des déchets nucléaires à vie longue en couche géologique profonde. Ce premier chapitre introductif est organisé comme suit. Dans le premier paragraphe, nous présentons le contexte général du projet de thèse. Le paragraphe suivant traite du contexte scientifique. La troisième partie introduit les modèles mathématiques qui seront utilisés dans la suite. Le quatrième paragraphe aborde l'estimation de paramètres à proprement dit. La dernière partie présente le travail réalisé sur l'identification de zones.

1.1 Contexte général

1.1.1 Le stockage des déchets nucléaires : enjeux, outils et difficultés

Depuis le début des années 1990, la protection de l'environnement est devenue une préoccupation collective. La question des déchets (au sens large) est quotidienne et touche chaque individu tant sur le plan professionnel que familial. Les enjeux liés à ces problèmes résident principalement dans la gestion des ressources en eau et la maîtrise des pollutions souterraines.

Parmi ces déchets, la problématique de la gestion des déchets nucléaires est devenue durant ces dernières années une préoccupation importante de la société. Il devient en effet nécessaire de trouver une solution durable au stockage des déchets nucléaires, résidus des activités civiles et militaires, afin de préserver notre environnement. Ces déchets ayant une longue durée de vie, il convient de leur trouver une solution appropriée respectant les contraintes de sûreté. L'objectif de la gestion des déchets est de protéger l'homme et son environnement contre toute émission ou dissémination de matières dangereuses.

En France, l'Andra* travaille depuis longtemps sur le sujet et a pour mission d'étudier, de concevoir, d'exploiter et de surveiller les centres de stockage des déchets radioactifs en protégeant à long terme l'homme et l'environnement de l'impact des déchets. L'Andra étudie notamment des solutions de gestion durable pour les déchets qui restent aujourd'hui sans filière industrielle de gestion. Pour ce faire, elle s'appuie sur les mathématiques appliquées afin de trouver les solutions les mieux adaptées. En effet, le seul moyen de prédire le comportement des déchets en sous-sol sur plusieurs milliers d'années est de réaliser des simulations numériques fiables du transport de radionucléides en milieu souterrain. Dans l'hypothèse où des radioéléments dangereux seraient relâchés dans la biosphère, on doit pouvoir disposer de toutes les informations nécessaires à la compréhension des circonstances du relâchement et d'en connaître les proportions.

Les principales difficultés liées au problème de stockage sont de plusieurs ordres :

- la première est d'ordre scientifique et technique : comment assurer la sûreté du stockage sur le long terme ? De plus, la récupération, le traitement et l'interprétation des données issues du laboratoire souterrain sont assez lourds. Les modèles sont complexes, les simulations (directes et inverses) demandent donc une grande capacité de calcul des ordinateurs et des méthodes numériques adaptées ;
- la deuxième est d'ordre financière : le coût de ce travail est énorme, tant en moyen humains (chercheurs, ingénieurs, techniciens), qu'en matériel (construction du laboratoire de Bure où les tests sont effectués, carottages, instruments de mesure, moyens

*Agence Nationale pour la gestion des Déchets RadioActifs
1/7 rue Jean Monnet, Parc de la Croix Blanche, 92298 Châtenay-Malabry.
Voir l'URL <http://www.andra.fr/>

informatiques). Il se pose également la question du coût du stockage qui est payé par l'ensemble de la société et qui ne doit pas nuire à sa sûreté.

1.1.2 Les différents types de déchets

La classification des déchets se fait par rapport aux caractéristiques des déchets en fonction de leur activité et de la durée de vie des radionucléides. Pour les différentes catégories de déchets, l'Andra étudie puis met en œuvre des solutions de gestion adaptées.

On distingue traditionnellement trois types de gestion pour les déchets HAVL correspondant aux trois axes de la loi de 1991 (cf. section 1.1.3) : l'entreposage en surface, la transmutation-incinération, le stockage en site profond. Ces trois voies sont actuellement explorées en France par l'Andra et le CEA. Le stockage en couche géologique profonde est la solution de référence pour les déchets à haute ou moyenne activité à vie longue (HA-VL ou MA-VL), voir notamment [86]. On parle de déchets à vie courte pour ceux contenant des isotopes avec une période radioactive (temps nécessaire pour qu'un élément radioactif perde la moitié de son activité par désintégration naturelle) inférieure à 30 ans, de déchets à vie longue pour les autres. Les caractéristiques de la couche d'argilites du Callovo-Oxfordien étudiée sont susceptibles de garantir la sûreté du stockage. L'Andra explore donc les propriétés de ces argilites dans son laboratoire souterrain de Meuse/Haute Marne construit au milieu de la couche du Callovo-Oxfordien à 500 m de profondeur (figure 1.1).

A l'inverse des déchets de très faible activité qui atteignent un niveau d'activité proche de la radioactivité naturelle en quelques dizaines d'années, les déchets de haute activité à vie longue nécessitent d'être confinés pendant des durées de l'ordre du million d'années avant de retrouver une radioactivité équivalente à celle du milieu naturel. Le stockage doit donc garantir une sûreté à très long terme. Ces déchets sont conditionnés en vue de leur entreposage puis de leur stockage, c'est-à-dire qu'ils sont solidifiés pour ceux qui se présentent sous forme dispersable, puis placés dans des conteneurs en béton ou en acier. Ces conteneurs, aussi appelés colis, varient selon les déchets auxquels ils sont destinés mais le principe général reste toujours le même : les déchets sont mis dans une matrice généralement composée de ciment et/ou de bitume.

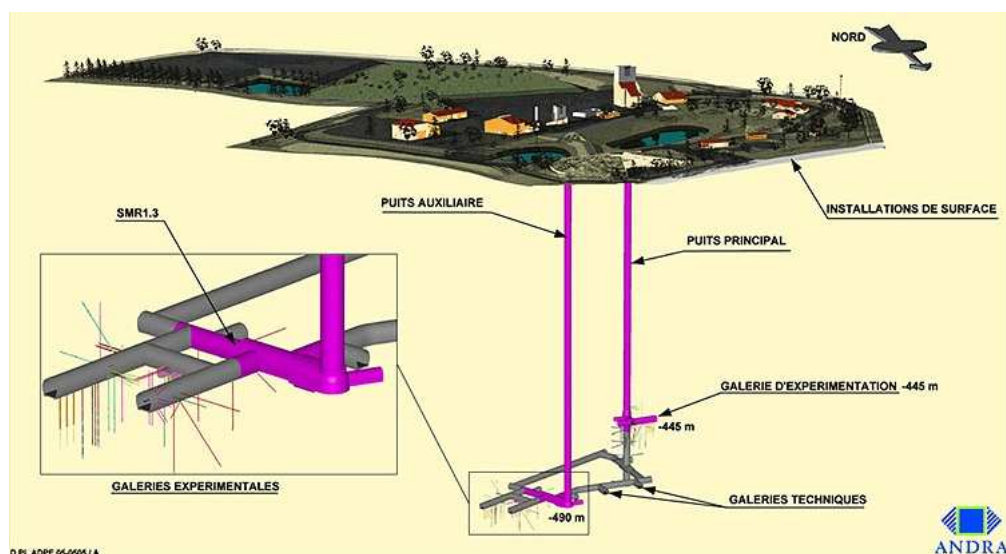


FIG. 1.1 – Laboratoire souterrain de Bure

1.1.3 Les lois de 1991 et 2006 et le rôle de l'Andra

Suite aux études menées par l'Andra dans le cadre de la loi de 1991 (dite loi "Bataille") et présentées dans le rapport remis fin 2005 au gouvernement, le parlement français a voté la loi n°2006-739 du 28 juin 2006 relative à la gestion durable des matières et des déchets radioactifs. Cette loi précise les recherches et les études relatives aux déchets de haute activité (HA) et de moyenne activité à vie longue (MA-VL) et renforce le positionnement de l'Andra comme responsable de la gestion durable de tous les déchets radioactifs produits par la France. Les principales évolutions des missions de l'Andra concernent :

- le pilotage des recherches sur l'entreposage,
- la conception, l'implantation et la gestion des entreposages et stockage,
- la prise en charge des déchets et des sites orphelins (mission de service public).

Le Dossier 2005 ([8]) conclut à la faisabilité du stockage géologique réversible dans les argilites similaires à celles du Callovo-Oxfordien. C'est sur la base des éléments de ce dossier, des conclusions du CEA sur les deux autres axes de recherche (la transmutation-incinération et l'entreposage de longue durée) et d'un débat public, qu'a été promulguée la loi de 2006 qui confirme les trois axes de recherche pour la gestion des déchets HA et MA-VL : la séparation et la transmutation des éléments radioactifs à vie longue, confié au CEA, le stockage réversible en couche géologique profonde et l'entreposage, confiés à l'Andra.

Le projet de loi précise les jalons des études. Pour la séparation et la transmutation, devront être arrêtées, en 2012, les perspectives industrielles liées aux recherches sur la quatrième génération de réacteurs. Pour le stockage réversible en couche géologique profonde, il est prévu de réunir en 2015 tous les éléments nécessaires à une autorisation avec la tenue d'un débat public en 2012. Quant au centre de stockage éventuel, sa date de mise en service est fixée à 2025, ce qui est compatible avec le calendrier de production des déchets à haute activité et à vie longue issus du cycle nucléaire français.

Le texte apporte également deux éléments essentiels dans des domaines non couverts par la loi de 1991. D'une part, il propose une véritable gestion nationale pour les déchets, mais aussi pour les matières radioactives, valorisables ou non, en instituant le plan national de gestion des déchets radioactifs. D'autre part, il donne un cadre législatif au démantèlement des installations nucléaires.

1.1.4 Le stockage en couche géologique profonde

Le concept de stockage étudié par l'Andra pour les déchets HAVL est basé sur le stockage dans les argilites du même type que celles du Callovo-Oxfordien situées au laboratoire souterrain de Meuse/Haute Marne entre 420 et 550 mètres de profondeur. Très peu perméable, la couche d'argile d'une épaisseur de 130 m présente des propriétés de rétention des radioéléments importantes. L'âge de la formation, sa stabilité tectonique, la profondeur d'implantation du stockage permettent d'envisager une grande stabilité de ces propriétés favorables, aux échelles de temps étudiées ici (de l'échelle du millier d'années à celle de plusieurs centaines de milliers d'années).

L'eau est à la fois le principal facteur de corrosion des colis de déchets radioactifs et du relâchement consécutif des radioéléments qu'ils contiennent, et le vecteur de leur transport dans la barrière géologique. Une fois dissous dans l'eau, les radioéléments seront transportés par diffusion/dispersion et convection à travers les couches géologiques. On voit donc que la couche d'argile a le double rôle de retarder le contact de l'eau avec les colis et la contamination de la biosphère (et donc de laisser le temps à de nombreux isotopes de

se désintégrer).

Dans ce contexte, le stockage doit être conçu pour s'opposer à la circulation d'eau, limiter le relâchement des radionucléides et les immobiliser à l'intérieur du stockage lui-même, retarder et atténuer la migration des radionucléides qui auraient été relâchés par les déchets et préserver les propriétés favorables du milieu en limitant les perturbations.

L'architecture du stockage envisagée se définit sur plusieurs niveaux. Le stockage se décompose en plusieurs zones destinées à recevoir différents types de déchets. Chaque zone est composée de modules qui sont reliés par des galeries de liaison et traversés par des galeries d'accès. Les modules sont constitués d'alvéoles (figure 1.2) connectées aux galeries d'accès et contiennent les colis à stocker. Les alvéoles sont des cavités souterraines creusées dans la formation géologique, leur architecture précise varie selon la zone de stockage pour s'adapter au type de déchets reçus et à leur conditionnement. La conception du stockage vise à minimiser l'extension verticale du stockage afin de pouvoir s'insérer dans la couche d'argile du Callovo-Oxfordien et de maximiser les épaisseurs d'argilites situées au-dessus et au-dessous du stockage.

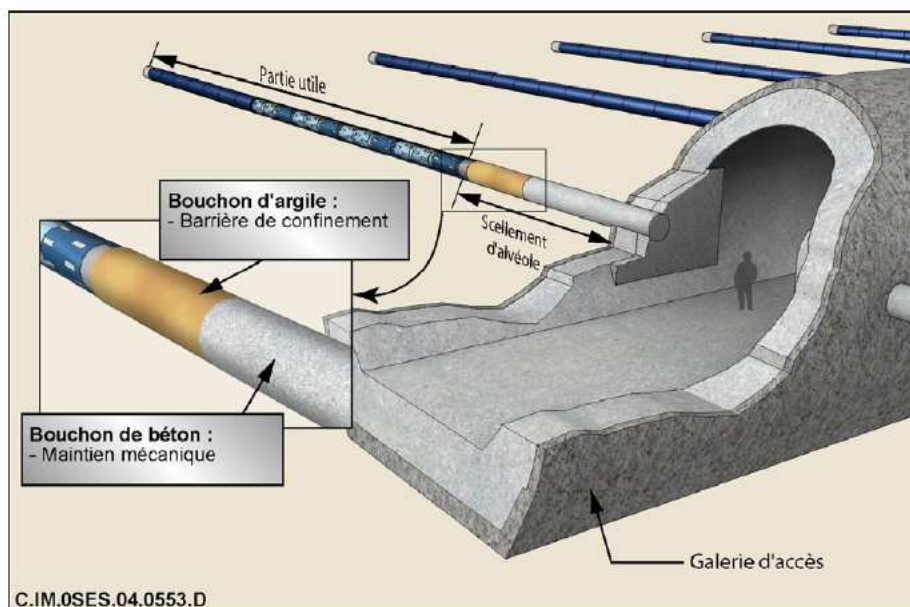


FIG. 1.2 – Structure d'un module

1.2 Contexte scientifique

Le travail de la thèse s'inscrit dans le contexte du projet de stockage en formation géologique profonde de déchets radioactifs haute et moyenne activité et vie longue (HAMAVL). L'objectif fondamental de la gestion à long terme de ces déchets est de protéger, sur une très grande durée, l'homme et l'environnement des risques de contamination. La simulation sur de grandes échelles d'espace et de temps, et notamment la simulation des écoulements et du transport des radionucléides est un outil important pour évaluer la sûreté et l'impact du stockage sur les personnes et leur environnement.

1.2.1 Problématiques liées au stockage des déchets à haute et moyenne activité et à vie longue

Le stockage des déchets nucléaires en site profond nécessite une bonne maîtrise des lois de diffusion/dispersion des radioéléments dans les couches géologiques. Pour cela, la modélisation mathématique doit prendre en compte certains mécanismes comme par exemple la dissolution/précipitation, la filiation des espèces, l'adsorption. Les méthodes numériques que nous utiliserons doivent donc être adaptées aux modèles physiques et mathématiques décrivant ces phénomènes.

La simulation numérique, à grande échelle et sur une longue période, du transport de radionucléides en couche géologique profonde pose de nombreux problèmes. On peut tout d'abord mentionner les difficultés liées aux hétérogénéités du domaine de calcul. Ces hétérogénéités existent à plusieurs échelles en espace. A l'échelle la plus grande, elles sont causées par les changements de couches géologiques. Cela correspond à des changements dans les propriétés hydrogéologiques des roches qui composent ces couches, provoquant, par exemple, des variations de perméabilité ou de coefficients de diffusion-dispersion de plusieurs ordres de grandeur. Ces hétérogénéités se traduisent par des discontinuités dans les coefficients des équations au passage d'une couche géologique à l'autre. Cependant, certaines quantités physiques restent continues : les débits en eau, la pression et les flux de contaminants.

La durée des simulations constitue une autre difficulté. En effet, celle-ci est en général longue car les écoulements souterrains et la diffusion des radionucléides sont lents (en comparaison avec ce qui se passe en surface ou dans l'air). Dans le cas des déchets à haute activité et à vie longue, la durée de simulation peut aller jusqu'à un million d'années. Cette durée est alors choisie de sorte que le maximum de l'impact potentiel du stockage sur l'environnement soit dépassé et puisse être évalué. Dans de telles échelles de temps, les colis seront corrodés et des radionucléides seront relâchés dans le stockage. Il est donc essentiel de pouvoir évaluer la propagation des radionucléides s'ils venaient à être libérés dans le milieu géologique. Ces résultats n'étant pas accessibles par des voies expérimentales, il est nécessaire de résoudre un modèle mathématique complexe que nous présenterons dans la section 1.3.

Notre étude doit aussi prendre en compte la zone endommagée lors du forage des galeries et des alvéoles de stockage et il faut donc caractériser les propriétés hydrauliques et de transport de ces zones et des matériaux (béton par exemple) utilisés pour construire les ouvrages d'entreposage. On peut effectivement s'attendre à avoir des valeurs de paramètres différentes dans cette zone par rapport à la zone saine. Enfin, on doit également mentionner les problèmes concernant les incertitudes sur les données, issues de forages à 500 m sous terre, et leur impact sur les résultats, ainsi que l'analyse de la sensibilité des résultats par rapport à nos paramètres.

1.2.2 Le laboratoire souterrain et l'expérimentation DIR

L'expérimentation DIR – Diffusion de traceurs Inertes et Réactifs – appartient au programme de recherche multidisciplinaire engagé par l'Andra pour étudier les caractéristiques et le comportement des argilites de la formation du Callovo-Oxfordien du site de Meuse/Haute Marne pour le stockage des déchets radioactifs. L'objectif de cette expérimentation est d'améliorer la compréhension des mécanismes de transfert des radionucléides dans les argilites et d'évaluer les valeurs de ses paramètres de diffusion qui sont nécessaires aux analyses de sûreté et aux études de faisabilité d'un stockage.

Le Callovo-Oxfordien est une couche sédimentaire d'argilites de 130 m d'épaisseur en-

viron. Cette roche est saturée en eau et sa porosité est comprise entre 10 et 20 %. Depuis 1994, les différentes campagnes d'expérimentation ont montré la faiblesse des flux convectifs au sein de la formation du Callovo-Oxfordien, et n'a pas mis en évidence de discontinuité susceptible d'être le siège d'écoulements significatifs. En conséquence, le principal facteur du transport des radionucléides est la diffusion moléculaire.

Le principe d'un essai de diffusion *in situ* est le suivant : des traceurs en solution sont mis en contact avec le milieu géologique à partir d'une chambre de test constituée d'une section de forage délimitée par des obturateurs. La concentration en traceurs dans la section d'essai est suivie sur une année environ.

Les traceurs dans l'expérience DIR sont de trois types :

- les traceurs inertes, comme l'eau tritiée, donnent accès aux seules propriétés de géométrie de la porosité de la roche et donc au coefficient de diffusion effectif,
- les traceurs anioniques, comme le chlorure, repoussé par les milieux argileux, donnent accès à l'effet conjoint de la diffusion et de la porosité accessible à la diffusion,
- les cations retenus par la roche donnent accès à l'effet conjoint de la diffusion et de la rétention chimique.

Concernant la récupération des données de l'expérimentation DIR, on peut préciser que la concentration est mesurée par l'analyse de l'eau contenue dans la roche ou dans la chambre d'injection. La pression est directement mesurable à l'aide d'un transducteur de pression. Si la porosité est aussi une quantité directement mesurable, les incertitudes sur la valeur de ce paramètre sont liées à sa variabilité spatiale : on ne peut en effet mesurer la porosité qu'en un nombre limité de points.

1.2.3 Problème inverse et mesures

Une problématique qui n'a pas encore été abordée est celle concernant les mesures. En effet, tous les paramètres de notre modèle mathématique (1.3) ne sont pas directement mesurables lors des expérimentations (ou alors des mesures sont disponibles mais en quantités insuffisantes), ce qui nécessite de mettre en œuvre des techniques de résolution des problèmes inverses afin de les identifier. Nous reviendrons plus en détails sur les problèmes inverses dans la section 1.4.

Dans cette thèse, nous avons utilisé des données synthétiques (*i.e.* auto simulées) afin de valider les méthodes employées. L'avantage d'utiliser des données synthétiques est que l'on peut comparer la porosité et le champ de diffusion obtenus en résolvant le problème inverse avec la porosité et le champ de diffusion initiaux (qui sont en pratique inconnus), construire des données perturbées aléatoirement et ainsi évaluer la précision des méthodes numériques mises en œuvre. Les données de concentration utilisées diffèrent pour chaque cas-test mis en œuvre et sont exposées dans le chapitre 5. Une des perspectives de notre travail est l'application des méthodes présentées dans ce document aux mesures relevées dans l'expérience DIR et qui consistent en des mesures de concentration recueillies dans la chambre d'injection tout au long de l'expérience, ainsi que des mesures au temps final obtenues par un sur-carottage (cf. chapitre 7).

1.3 Modèles physiques et mathématiques

Dans cette thèse, nous avons considéré les problèmes liés au transport de radionucléides en milieu poreux. Pour modéliser ce type de phénomène, nous utilisons un modèle de convection-diffusion/dispersion en milieu poreux non déformable et peu perméable (écoulement Darcéen). Le principal phénomène de transport dans le Callovo-Oxfordien étant la dif-

fusion moléculaire, nous définirons les paramètres nécessaires pour décrire ce phénomène : le coefficient de diffusion, noté D ; la porosité accessible à la diffusion, ω et le coefficient de retard R , égal à 1 dans le cas des traceurs non retardés chimiquement.

Nous évoquerons également les équations d'écoulement même si notre étude porte sur le transport. En effet, les modèles d'écoulements servent de base aux modèles de transport, il nous semble donc nécessaire de les mentionner afin d'avoir une bonne compréhension de tous les phénomènes en jeu. Une littérature abondante existe sur les problèmes en milieu poreux, on peut citer par exemple Alboin et al. [2, 3], Apoung et al. [9], Chavent et al. [38], de Dieuveult et al. [44], Jaffré [63, 64].

Dans toute la suite, on notera Ω le domaine d'étude.

1.3.1 Les écoulements en milieu poreux saturé

Définition 1.1 (*Milieu poreux*) *Un milieu poreux est un milieu composé d'une matrice solide et d'espaces vides (les pores) pouvant être occupés par de l'air, de l'eau ou d'autres fluides. On peut caractériser un milieu poreux par deux grandeurs : sa porosité et sa perméabilité.*

Dans notre contexte, on s'intéressera uniquement au cas d'un milieu saturé en eau, c'est-à-dire dans lesquels tous les vides sont remplis d'eau. Les sols peuvent donc être assimilés à des milieux poreux contenant une certaine proportion d'eau. La migration des espèces chimiques dans les sols se fait par le biais de l'eau qu'ils contiennent : les espèces chimiques sont dissoutes dans l'eau et les sols perméables permettent la circulation de l'eau et des espèces. On peut distinguer deux types de contributions essentielles dans la migration d'une espèce : d'une part un mouvement d'ensemble porté par la circulation de l'eau dans le milieu (convection) et d'autre part une tendance à homogénéiser la concentration. Cette tendance provient de deux mécanismes distincts : la dispersion cinématique et la diffusion moléculaire. Selon la nature des espèces et du milieu, d'autres phénomènes peuvent encore influencer la migration.

Par la suite, la roche poreuse est observée à une échelle telle que les pores sont très petits par rapport à l'échelle retenue, ce qui permet de considérer le milieu comme continu.

Définition 1.2 (*Porosité*) *La plupart des roches contiennent naturellement un certain pourcentage de vide, qui peut être occupé par de l'eau ou des polluants : c'est ce qu'on appelle la porosité. La porosité ω est constituée par les espaces libres qui existent entre les grains de la roche ou par les fissures créées par les mouvements tectoniques de la croûte terrestre. La porosité mesure donc le rapport de volume entre la roche et le vide.*

Pour caractériser la porosité d'un milieu géologique, il convient de définir un volume élémentaire représentatif dont la taille permet de déterminer une valeur moyenne de la porosité telle que les effets dus à la variation de la taille des pores soient négligeables. La taille du volume représentatif doit également garantir que les variations des paramètres d'un volume à un autre peuvent être approchées par des fonctions continues et correspondre aux échelles de dimension des instruments mesurant les valeurs des paramètres. Ce concept de volume représentatif permet de représenter le domaine géologique par un milieu poreux équivalent maillé par "un certain nombre de points" pour lesquels les propriétés physiques du milieu sont approchées par des grandeurs moyennes traduisant, à l'échelle du volume entourant chaque point, les hétérogénéités du milieu géologique.

Définition 1.3 (*Perméabilité*) *La notion de perméabilité k quantifie l'influence de la géométrie de la matrice solide sur la vitesse d'écoulement des fluides dans la roche. Elle dépend*

de la porosité ω , des liaisons entre les pores et de la nature des fluides. Dans le cas général, k est un tenseur.

Définition 1.4 (*Conductivité hydraulique*) La conductivité hydraulique K du milieu poreux est définie comme le rapport entre la perméabilité et la viscosité de l'eau. Selon la loi de Darcy, le tenseur de conductivité hydraulique est le rapport entre le flux volumique d'écoulement de l'eau, aussi appelé vitesse de Darcy q , et le gradient en pression.

On considère que le milieu est hétérogène, *i.e.* les valeurs de porosité, perméabilité et diffusion ne sont pas constantes. Le modèle mathématique de l'écoulement est le suivant

$$\begin{cases} s \frac{\partial h}{\partial t} + \nabla \cdot q = f \\ q = -K \cdot \nabla h \end{cases} \quad (1.1)$$

où s est un coefficient de stockage spécifique, $h(x, t)$ la charge piézométrique, $f(x, t)$ est un terme source et K dénote le tenseur de conductivité hydraulique qui s'écrit sous la forme

$$K = \frac{k \rho g}{\mu} \quad (1.2)$$

avec k le tenseur de perméabilité, g la gravité, ρ la densité du fluide et μ la viscosité du fluide.

La première équation exprime la conservation de la masse d'eau. Comme nous considérons un écoulement incompressible, la densité volumique du fluide est constante dans le temps et dans l'espace. La deuxième équation est appelée loi de Darcy. Elle traduit le fait que la vitesse de Darcy est proportionnelle au gradient de pression. Celle-ci n'est valable que sous certaines conditions : de faibles vitesses (permettant de négliger l'inertie du fluide) et des milieux qui ne sont pas trop hétérogènes.

Comme l'équation d'écoulement est indépendante de la concentration, on peut la traiter en premier et résoudre ensuite de façon séparée l'équation de transport. Pour certains contaminants, il est possible que la viscosité dépende de la concentration, mais on suppose généralement que les radionucléides sont présents en quantités suffisamment petites pour que l'influence de leur concentration sur la viscosité soit négligeable.

1.3.2 Le transport de radionucléides en milieu poreux saturé

Pour décrire le transport d'espèces chimiques en milieu poreux, il convient de définir quelques mécanismes de base et ceux spécifiques à la propagation d'espèces chimiques telles que les radionucléides.

Définition 1.5 (*Convection ou advection*) La convection est le mécanisme le plus intuitif de déplacement, il consiste en l'entraînement des éléments en solution par le fluide qui se déplace.

Définition 1.6 (*Dispersion cinématique*) La variation des vitesses en amplitude et en direction à petite échelle, due à la microstructure du milieu poreux, entraîne une dilution du soluté appelée dispersion cinématique.

Définition 1.7 (*Diffusion moléculaire*) La diffusion moléculaire est un phénomène physique lié à l'agitation moléculaire. Contrairement à la dispersion cinématique, il peut être indépendant de la convection.

Définition 1.8 (*Sorption*) *Le cas idéal de la convection-diffusion pure est rarement rencontré en pratique et il est par exemple fréquent que le soluté interagisse avec la matrice du milieu poreux : on parle alors de sorption. il existe différents types de sorption, on peut par exemple citer l'adsorption qui consiste en l'accumulation d'éléments sur l'interface entre la solution et le solide.*

Définition 1.9 (*Décroissance radioactive*) *La décroissance radioactive est la réduction du nombre de noyaux radioactifs (instables) dans un échantillon. La décroissance radioactive se produit jusqu'à ce que tous les noyaux de l'échantillon soient stables.*

Définition 1.10 (*Filiation*) *La filiation exprime le fait qu'un radionucléide père peut former en se désintégrant d'autres radionucléides moins instables.*

Les phénomènes tels que la propagation de concentration en milieu poreux sont décrits par des équations aux dérivées partielles qui relient les paramètres (porosité, diffusion) à la variable d'état (concentration). Le modèle mathématique complet exprime la conservation de la masse de chaque radionucléide et se compose d'une équation pour la partie dissoute (1.3) et d'une équation pour la partie précipitée (1.4).

Le transport du radionucléide i en phase liquide est modélisé par l'équation de bilan suivante

$$\frac{\partial(\omega R_i C_i)}{\partial t} = \nabla \cdot (D_i \cdot \nabla C_i - q C_i) - \lambda_i \omega R_i C_i + \sum_j \sigma_{ij} \lambda_j \omega R_j C_j + \omega S_i + Q_i \quad (1.3)$$

où $\omega(x)$ est la porosité du milieu, R_i le coefficient de retard du radionucléide i , $C_i(x, t)$ la concentration du radionucléide i dissout dans l'eau, $D_i(x)$ le tenseur de dispersion/diffusion, $q(x)$ désigne la vitesse de Darcy, λ_i le coefficient de décroissance radioactif associé au radionucléide i , j décrit l'ensemble des pères du radionucléide i , σ_{ij} est la fraction du radionucléide j se désintégrant en le radionucléide i , $S_i(x, t)$ est le terme d'échange entre les phases solides et liquides et $Q_i(x, t)$ le terme source, dépendant du contenu et de l'état de la barrière ouvragée.

L'équation de transport (1.3) contient

- ★ un terme d'accumulation noté $\omega R_i \frac{\partial C_i}{\partial t}$,
- ★ un terme de diffusion-dispersion $\nabla \cdot (D_i \cdot \nabla C_i)$, un terme d'advection $\nabla \cdot (q C_i)$,
- ★ un terme de décroissance radioactive $\lambda_i \omega R_i C_i$, un terme de filiation $\sum_j \sigma_{ij} \lambda_j \omega R_j C_j$.

L'équation de bilan en phase solide s'écrit

$$\frac{\partial[(1 - \omega) f_i]}{\partial t} + \lambda_i (1 - \omega) f_i = \sum_j \sigma_{ij} \lambda_j (1 - \omega) f_j - \omega S_i \quad (1.4)$$

où f_i est la concentration de radionucléide i précipitée dans la matrice rocheuse.

La loi d'échange entre les phases liquides et solides est donnée par

$$S_i = \varrho_i (C_i^{\text{sat}} - C_i) \delta_i \quad \text{avec} \quad \delta_i = \begin{cases} 0 & \text{si } f_i = 0 \text{ et } C_i < C_i^{\text{sat}} \\ 1 & \text{sinon} \end{cases} \quad (1.5)$$

où ϱ_i est la constante cinétique de précipitation/dissolution et C_i^{sat} la concentration de saturation du radionucléide i . La loi d'échange entre les deux phases (1.5) exprime la

prise en compte de la précipitation par isotope. On peut en effet constater pour certains contaminants un phénomène de **précipitation** : la concentration dissoute ne peut pas dépasser un certain seuil C_i^{sat} au-delà duquel les radionucléides en surnombre précipitent, c'est-à-dire qu'ils se fixent sur la matrice rocheuse sous forme solide. A l'endroit où un contaminant a précipité, sa concentration en phase liquide est localement constante, égale à la concentration de saturation C_i^{sat} . L'échange entre la phase liquide et la phase solide est quantifié par un coefficient d'échange constant pour un radionucléide donné.

Remarque 1.1 *La loi d'échange (1.5) n'est pas différentiable par rapport aux concentrations aux points où $f_i = 0$ et $C_i = C_i^{\text{sat}}$.*

Dans la suite, nous utiliserons **un modèle simplifié** : on ne tiendra pas compte de la filiation, c'est-à-dire que le terme $\sum_j \sigma_{ij} \lambda_j \omega R_j C_j$ disparaît et donc les équations pour les différents types de radionucléides seront indépendantes. On considérera également des radionucléides qui ne précipitent pas. Si l'on ne considère qu'un seul radionucléide, l'équation (1.3) devient alors

$$\frac{\partial(\omega RC)}{\partial t} - \nabla \cdot (D \cdot \nabla C - q C) + \lambda \omega RC - Q = 0. \quad (1.6)$$

Le coefficient de diffusion D s'exprime comme la somme d'un terme de diffusion moléculaire effectif D_e dépendant de la porosité ω et du coefficient de diffusion moléculaire D_m et d'un terme D^* dépendant de la vitesse de Darcy

$$D = D_e + D^*. \quad (1.7)$$

Le coefficient D_e s'écrit sous la forme $D_e = \omega D_m$ et le terme D^* est une matrice, plus précisément c'est un tenseur symétrique du second ordre, qui s'écrit en deux dimensions sous la forme

$$D^* = \begin{pmatrix} \alpha_T \|q\| + (\alpha_L - \alpha_T) \frac{q_x^2}{\|q\|} & (\alpha_L - \alpha_T) \frac{q_x q_y}{\|q\|} \\ (\alpha_L - \alpha_T) \frac{q_x q_y}{\|q\|} & \alpha_T \|q\| + (\alpha_L - \alpha_T) \frac{q_y^2}{\|q\|} \end{pmatrix} \quad (1.8)$$

où α_L est la dispersivité longitudinale et α_T la dispersivité transversale.

Remarque 1.2 *Le calcul préalable de la vitesse de Darcy q est nécessaire pour pouvoir résoudre l'équation de transport, i.e. nous avons à résoudre (1.1) dans Ω avec des conditions initiales et aux limites appropriées avant de résoudre (1.6).*

1.4 Présentation du problème inverse

Certains coefficients, appelés aussi paramètres, intervenant dans un modèle mathématique sont inaccessibles expérimentalement. On parle d'identification, ou d'estimation de paramètres, lorsque l'on cherche à déterminer la valeur de ces paramètres à partir des mesures et des sorties d'un modèle mathématique : c'est ce qu'on appelle résoudre un *problème inverse*.

1.4.1 Estimation des paramètres de transport

La première partie de notre travail est d’estimer les paramètres (coefficients) de transport de l’équation de convection-diffusion (1.6), c’est-à-dire le coefficient de diffusion avec son anisotropie et la porosité accessible à la diffusion.

Un des objectifs de la thèse consiste donc à développer un outil numérique d’estimation de paramètres (problème inverse) de transport à l’échelle métrique. Ce travail nécessite dans un premier temps de réaliser une étude bibliographique des méthodes de résolutions applicables pour l’identification des paramètres hydrauliques et de transport d’un milieu poreux. L’accent sera mis en particulier sur les différentes méthodes d’optimisation dans le cadre de méthodes déterministes. Dans un second temps, il s’agit de mettre en œuvre l’outil proprement dit de résolution de problèmes inverses. Le milieu géologique pourrait présenter une zone endommagée avec variation des paramètres de porosité et de diffusion, il faudra donc en tenir compte dans la modélisation du domaine.

Nous formulerons tout d’abord le problème inverse de façon théorique en écrivant l’équation adjointe de l’équation de transport. La spécificité du travail sera d’utiliser la technique de zonation qui consiste à considérer le milieu comme un ensemble de zones homogènes, pour chacune desquelles les coefficients seront estimés*. L’approche choisie pourra être appliquée à des expériences dans des milieux dont l’échelle d’espace peut être très différente. La zonation, dans le cas de l’expérience DIR, devra prendre en compte la zone endommagée par le forage. La démarche adoptée consistera en premier lieu à identifier les coefficients dans des zones fixées à partir de mesures de concentrations. Pour cela, on a développé le code adjoint du code de diffusion-convection.

1.4.2 Identification de zones

Après avoir considéré que la structure du milieu était connue dans la partie I (estimation des paramètres), la seconde partie du travail consiste à essayer d’identifier cette structure, c’est-à-dire de définir des zones du domaine où les paramètres de transport identifiés peuvent être considérés comme constants. Pour cela, on essaye de déplacer les zones et éventuellement d’en créer de nouvelles. Cette approche est particulièrement intéressante (tant sur le plan scientifique que sur le plan technique) car nous sommes confrontés à ce type de configuration dans la pratique. Nous essayerons donc dans le chapitre 6, partant d’une configuration géologique inconnue, d’arriver à retrouver les zones ainsi que les paramètres associés à cette zone.

Les outils développés dans la partie identification de paramètres sont réutilisés ici. En pratique, nous identifierons les coefficients dans chaque maille ce qui nous permettra, en regroupant les mailles où les paramètres peuvent être considérés comme égaux, de définir des zones et de vérifier qu’elles permettent d’obtenir une solution cohérente avec les “mesures”. Comme cette méthode requiert une valeur de paramètre par maille, nous avons dû développer un préprocesseur pour passer de données par zones à des données par maille.

1.4.3 Aspect implémentation

Le code de résolution du problème de transport qui sert de base à ce travail est le logiciel Traces [62] (Transport RéActif de Contaminant dans les Eaux Souterraines) écrit essentiellement en fortran 90. Nous utiliserons également la plate-forme de simulation Alliances

*Il s’agit là d’un choix de représentation du domaine d’étude. Il n’est pas lié à la formulation théorique du problème inverse.

dont le langage de programmation est en python. La plate-forme Alliances, développée par l'Andra en partenariat avec le CEA et EDF, permet d'exploiter pour la simulation des écoulements et du transport en milieu poreux, trois codes de calcul : Porflow, Cast3m et Traces (cf. chapitre 2, section 2.3). Notre choix s'est porté sur Traces car ce code développé par l'Institut de Mécanique des Fluides et des Solides (IMFS) à Strasbourg et modifié par l'Andra dans le cadre du projet Alliances est adapté à notre type de problème et disponible en code source.

Il a cependant fallu complètement réadapter ce code afin de pouvoir dans un premier temps le différentier correctement avec le logiciel de différentiation automatique Tapede [92] développé à l'INRIA Sophia-Antipolis par l'équipe Tropics. Tapede est un générateur automatique de code que nous présenterons plus longuement au chapitre 4. Les changements et les ajouts opérés dans le code sont nombreux : réécriture et transformation de certaines parties, implémentation de la fonction coût du problème inverse, création de fichiers de données en binaire pour ne pas avoir d'erreur d'arrondis, programmation des routines et des interfaces de résolution du problème d'estimation de paramètres et d'identification de zones, etc. Dans un second temps, nous avons ajouté des morceaux de code afin de faire le lien entre les différents paramètres du programme : fonction coût, gradient et minimiseur. A cet effet, nous avons intégré au code le minimiseur N2QN1 [69] de telle sorte qu'une fois la fonction coût calculée par Traces et le gradient calculé avec Tapede, ceux-ci sont directement injectés dans N2QN1 qui commence son processus de minimisation. Un des avantages du code ainsi remanié est sa capacité à traiter aussi bien le problème d'estimation de paramètres que le problème d'identification de zones.

Nous avons également mis au point une interface avec Matlab permettant de traiter les résultats et de récupérer des images et des films de bonne qualité.

1.5 Plan du mémoire

La première partie de ce mémoire est consacrée au problème direct du transport de radionucléides en milieu poreux. Nous présentons (Chapitre 2) les modèles mathématiques ainsi que les différentes méthodes numériques employées. Nous introduisons également le logiciel Traces et la plate-forme Alliances sur lesquels les calculs ont été faits et nous exposons les résultats des simulations. La seconde partie du travail est ensuite destinée à l'estimation de paramètres. Le Chapitre 3 est dédié à l'étude de l'estimation des paramètres. Nous rappelons tout d'abord les principes de base du contrôle optimal, puis nous expliquons en détails la résolution théorique et numérique de notre problème inverse. Le Chapitre 4 présente les différentes techniques de calcul des dérivées et l'outil de différentiation automatique Tapede que nous avons utilisé pour calculer le gradient de la fonction coût. Le Chapitre 5 traite les résultats numériques obtenus dans cette partie sur des applications aux problèmes de l'Andra.

Nous nous penchons ensuite dans la troisième partie sur l'identification de zones. Dans le Chapitre 6, nous rappelons dans un premier temps les méthodes existantes pour l'identification de zones et dans un deuxième temps nous présentons la résolution du problème de zonation sur les applications définies au Chapitre 2. Enfin, le Chapitre 7 fait la synthèse du travail accompli et ouvre des perspectives de recherche.

Chapitre 2

Modèles mathématiques et numériques

Sommaire

2.1	Les équations d'état	16
2.1.1	L'équation d'écoulement	16
2.1.2	L'équation de transport	17
2.2	Discrétisation des équations	17
2.2.1	Introduction	17
2.2.1.1	Éléments finis mixtes hybrides	17
2.2.1.2	Éléments finis discontinus de Galerkin	18
2.2.2	Discrétisation de l'équation d'écoulement	18
2.2.3	Discrétisation de l'équation de transport : le code Traces	20
2.3	Plateforme de calcul Alliances	23
2.3.1	Définition & Historique	23
2.3.2	Fonctionnement de la plate-forme	23
2.3.3	Maillages & Post-traitement	24
2.4	Implémentation	25
2.4.1	Le code Traces	25
2.4.2	La plate-forme Alliances	26
2.5	Position du problème	26
2.5.1	Cadres et objectifs de notre travail	27
2.5.2	Présentation des cas-tests	27
2.5.2.1	Cas-test "colis"	28
2.5.2.2	Cas-test "réel"	29
2.6	Résolution du problème	30
2.6.1	Cas-test "colis"	30
2.6.2	Cas-test "réel"	31

Avant de présenter la résolution numérique des équations d'écoulement et de diffusion-convection, il convient de définir ce qu'on appelle par problème direct : on dit résoudre un **problème direct** (par opposition au **problème inverse**, cf. section 3.1) lorsqu'on cherche à déterminer l'évolution du champ de concentration dans l'équation (1.6) étant donnés les conditions initiales et aux limites ainsi que les paramètres de porosité ω et de diffusion D . Cela revient donc à résoudre un problème au sens "classique".

Dans le chapitre 1, nous avons présenté le modèle mathématique des équations de transport de radionucléides et d'écoulement en milieu poreux saturé. Nous revenons dans ce chapitre sur ces équations et nous présentons les modèles numériques associés. Nous n'écrivons pas ici les formulations variationnelles de ces équations, ce n'est en effet pas le but de notre étude.

Le chapitre est organisé comme suit : nous rappelons tout d'abord les équations d'état puis nous introduisons les schémas numériques qui ont servi à la discrétisation. Ensuite, nous expliquons le fonctionnement de la plate-forme Alliances et du code Traces et nous finissons par présenter le problème que nous considérons et exposer les résultats numériques du problème direct.

2.1 Les équations d'état

2.1.1 L'équation d'écoulement

Le but de ce paragraphe est de présenter succinctement l'équation d'écoulement afin d'avoir une bonne compréhension de l'équation de transport. On s'intéresse à un écoulement dans un milieu saturé en eau, c'est-à-dire dans lequel tous les vides sont remplis d'eau. Le modèle mathématique de l'écoulement est le suivant

$$\left\{ \begin{array}{l} s \frac{\partial h}{\partial t} + \nabla \cdot q = f \\ q = -K \cdot \nabla h \end{array} \right. \quad (2.1)$$

$$(2.2)$$

où s est un coefficient de stockage spécifique, $h(x, t)$ la charge piézométrique, $f(x, t)$ est un terme source et K dénote le tenseur de conductivité hydraulique. L'équation (2.1) exprime la conservation de la masse d'eau et la deuxième équation (2.2) représente la loi de Darcy. En deux dimensions, le tenseur K est de la forme

$$K = \begin{pmatrix} K_{xx} & K_{xy} \\ K_{xy} & K_{yy} \end{pmatrix}. \quad (2.3)$$

Etant donné que nous considérons un problème d'hydraulique saturé en régime permanent, il suffit de calculer une fois le champ de vitesse de Darcy avant de résoudre le problème de transport seul. En pratique, on se donnera (cf. résultats numériques du chapitre 5) directement un champ de vitesse pour l'écoulement plutôt que de résoudre l'équation de Darcy munie de conditions aux limites en charge hydraulique.

Remarque 2.1 *En hydraulique transitoire, la vitesse de Darcy couple l'hydraulique et le transport.*

2.1.2 L'équation de transport

Les contaminants s'échappant des colis sont dissous dans l'eau. Leur déplacement dans la barrière ouvragée puis la barrière géologique est dû d'une part au déplacement global de l'eau, quantifié par la vitesse de Darcy (du point de vue des contaminants, c'est la vitesse de convection) et d'autre part au phénomène de diffusion qui exprime la tendance des concentrations dissoutes à devenir homogènes. Le rapport entre la vitesse de diffusion des contaminants et le gradient de concentration est le tenseur de diffusion, qui est la somme d'un tenseur de diffusion moléculaire (scalaire) dépendant de la nature du contaminant considéré et d'un tenseur de diffusion (plein) dépendant de la vitesse de Darcy. Le transport d'un polluant dépend en premier lieu des caractéristiques de l'écoulement qui le transporte. Une approximation précise du champ de vitesse est donc nécessaire.

Comme annoncé dans l'introduction, nous utiliserons un modèle simplifié* : on ne tiendra pas compte de la filiation, les équations pour les différents types de radionucléides seront donc indépendantes. On considérera également des radionucléides qui ne précipitent pas (par exemple l'Iode 129). Si l'on ne considère qu'un seul radionucléide, l'équation a la forme suivante

$$\frac{\partial(\omega RC)}{\partial t} - \nabla \cdot (D \cdot \nabla C - q C) + \lambda \omega RC - Q = 0. \quad (2.4)$$

Le tenseur de diffusion D s'exprime comme la somme d'un terme de diffusion moléculaire effectif D_e et d'un terme D^* dépendant de la vitesse de Darcy

$$D = D_e + D^* \quad (2.5)$$

avec $D_e = \omega D_m$, D_m est le coefficient de diffusion moléculaire et D^* s'écrit en deux dimensions sous la forme

$$D^* = \begin{pmatrix} \alpha_T \|q\| + (\alpha_L - \alpha_T) \frac{q_x^2}{\|q\|} & (\alpha_L - \alpha_T) \frac{q_x q_y}{\|q\|} \\ (\alpha_L - \alpha_T) \frac{q_x q_y}{\|q\|} & \alpha_T \|q\| + (\alpha_L - \alpha_T) \frac{q_y^2}{\|q\|} \end{pmatrix} \quad (2.6)$$

avec α_L la dispersivité longitudinale et α_T la dispersivité transversale. Dans les applications numériques, on ne prendra pas en compte l'influence de la vitesse de Darcy *i.e.* la matrice D^* sera nulle. La matrice de diffusion sera donc réduite au coefficient de diffusion moléculaire effectif.

2.2 Discrétisation des équations

2.2.1 Introduction

2.2.1.1 Éléments finis mixtes hybrides

L'idée de base des méthodes mixtes (EFM) est d'approcher simultanément la fonction d'état et son gradient. Elles ont été présentées pour la première fois par Meissner [78].

*Les résultats de la thèse s'appliquent sans problème au modèle prenant en compte la filiation.

L'élaboration mathématique et numérique a été réalisée par Raviart & Thomas [83] et Thomas [95].

Avec la méthode des EFM associée à un espace de Raviart-Thomas de plus bas degré, on aboutit à un système matriciel symétrique mais non-défini positif. Pour remédier à cet inconvénient, une nouvelle formulation a été introduite : la méthode des éléments finis mixtes hybrides (EFMH). Le système matriciel est transformé en un système matriciel symétrique défini positif dont les inconnues sont des multiplicateurs de Lagrange. Ils sont introduits pour réaliser l'hybridation et s'interprètent comme des traces de la fonction d'état sur les arêtes (resp. les faces) des éléments en 2D (resp. en 3D). Pour plus de précisions, on peut se référer aux travaux de Nédélec [79], Brezzi & Fortin [24] et Arnold & Brezzi [10], à Farhloul & Serghini Mounim [49] pour une formulation EFMH des équations de convection-diffusion et enfin à Roberts & Thomas [85] pour une vision globale de la question.

2.2.1.2 Éléments finis discontinus de Galerkin

En 1973, Reed & Hill [84] introduisent la première méthode discontinue de Galerkin (EFDG) pour les équations hyperboliques. Depuis cette date, il y a eu un développement constant des méthodes de Galerkin qui a permis la mise au point d'un grand nombre de méthodes pour les problèmes hyperboliques et assimilés. Dans les années 1970, et indépendamment des travaux de Reed & Hill, des méthodes de Galerkin utilisant des éléments finis discontinus ainsi que de nombreuses variantes furent proposées et étudiées pour la résolution des équations paraboliques et elliptiques. Ces méthodes sont généralement appelées *interior penalty methods* (IPM) et leur développement est resté indépendant des méthodes de Galerkin pour les équations hyperboliques.

Concernant les EFDG, on peut mentionner les travaux de Arnold et al [11], Brezzi et al [25], le livre de Thomée [96] et le papier de Cockburn et al [42] évoquant l'évolution des méthodes EFDG et leur interaction avec les volumes finis.

Pour les équations d'advection-diffusion, on peut se référer à Siegel et al. [88] pour une solution utilisant une combinaison d'éléments finis discontinus et d'éléments finis mixtes.

2.2.2 Discrétisation de l'équation d'écoulement

L'équation du flux a été discrétisée à l'aide des EFMH. Elle est donnée naturellement en formulation mixte : elle couple l'équation de la conservation de masse (2.1) avec la loi de Darcy (2.2). La formulation mixte est particulièrement appropriée pour ce modèle, car elle donne une approximation directe de la vitesse de Darcy. En effet, notre intérêt n'est pas dans la charge h mais dans la concentration C , et dans l'équation du transport qui donne la concentration, c'est la vitesse de Darcy q qui intervient et non pas la charge. La méthode des différences finies et les méthodes classiques d'éléments finis donnent, en premier lieu, une approximation de la charge qui est dans un second temps différenciée pour donner l'approximation de la vitesse de Darcy qui est utilisée dans l'équation du transport. Cette différenciation est une source d'erreur et la précision est réduite. La méthode d'éléments finis mixtes-hybrides produit non seulement l'approximation de la pression et de la vitesse mais aussi une approximation de la trace de pression sur les interfaces du maillage. Grâce à cette méthode, nous disposons de la trace de la pression aussi bien que la trace normale de la vitesse sur les interfaces entre les sous-domaines.

Pour résoudre le problème d'écoulement, nous utilisons (comme pour le problème de transport) le code de calcul Traces [62] que nous présentons dans la section 2.2.3.

Schémas numériques

Les EFMH utilisés dans le code de calcul Traces sont définis par les fonctions de base de l'espace Raviart-Thomas de plus bas degré (ordre 0). Pour un élément E de nb côtés, les fonctions de base w_i sont définies de manière générale par

$$\int_{E_j} w_i \cdot n_{E_j} dx = \delta_{ij}, \quad j = 1, \dots, nb \quad (2.7)$$

où E_j est le côté de l'élément E , n_{E_j} est le vecteur sortant normal à E_j . On définit par q_E la vitesse de Darcy dans l'élément E et on a

$$q_E(x) = \sum_{j=1}^{nb} Q_{E_j} w_j \quad (2.8)$$

où Q_{E_j} représente le flux à travers chaque côté de l'élément E , les flux sortants étant de signe positif. On discrétise maintenant la loi de Darcy et l'équation de conservation séparément et ensuite par substitution, on construit un système d'équations ayant la charge moyenne par côté pour inconnues principales.

La loi de Darcy est écrite de la manière suivante

$$K_E^{-1} \cdot q_E = -\nabla h, \quad \text{avec } K_E = \begin{pmatrix} K_x & K_{xy} \\ K_{xy} & K_y \end{pmatrix} \quad (2.9)$$

où K_E représente le tenseur de conductivité hydraulique (en 2D). On écrit ensuite (2.9) sous forme variationnelle

$$\int_E (K_E^{-1} q_E) \cdot w_i dx = - \int_E \nabla h \cdot w_i dx \quad (2.10)$$

et en utilisant les propriétés des fonctions de base, (2.10) devient

$$\sum_{j=1}^{nb} Q_{E_j} \left(\int_E w_i \cdot (K_E^{-1} \cdot w_j) dx \right) = h_E - Th_{E_i} \quad (2.11)$$

où h_E est la charge moyenne dans la maille E et Th_{E_i} la charge moyenne sur le côté i . On définit la matrice B de la façon suivante

$$B_{E_{ij}} = \int_E w_i \cdot (K_E^{-1} \cdot w_j) dx. \quad (2.12)$$

On peut alors réécrire (2.11) sous forme matricielle

$$\sum_{j=1}^{nb} B_{E_{ij}} Q_{E_j} = h_E - Th_{E_i}. \quad (2.13)$$

Les flux à travers chaque côté sont donnés par

$$Q_{E_i} = \sum_{j=1}^{nb} B_{E_{ij}}^{-1} (h_E - Th_{E_i}). \quad (2.14)$$

L'équation de conservation de la masse (2.1) est écrite sous forme intégrale

$$\int_E \left(s \frac{\partial h}{\partial t} - \nabla \cdot q - f \right) dE = 0, \quad (2.15)$$

$$\text{soit } |E| s_E \frac{\partial h_E}{\partial t} - \sum_{j=1}^{nb} Q_{E_j} - F_E^S = 0 \quad (2.16)$$

où $|E|$ est l'aire de l'élément E , s_E est l'approximation de s et F_E^S le terme source associé à l'élément E .

La continuité de la charge h et des flux entre deux éléments adjacents E et E' est assurée par

$$Th_{E_i} = Th_{E'_j} \quad (2.17)$$

$$Q_{E_i} + Q_{E'_j} = 0. \quad (2.18)$$

Pour terminer le schéma en espace, il reste à réaliser l'**hybridation** des équations. Les différentes étapes de substitution sont les suivantes :

1. les flux dans l'équation de conservation (2.1) sont remplacés par leur expression obtenue lors de la discrétisation de la loi de Darcy (2.9),
2. la charge moyenne h_E dans l'élément est exprimée en fonction des charges moyennes sur les côtés Th_{E_i} (2.11) à partir de l'équation de conservation,
3. les flux Q_{E_j} sont remplacés par leur expression en charge (2.14) obtenue lors de la discrétisation de la loi de Darcy dans l'équation de continuité des flux (2.18),
4. l'équation en charge moyenne par côté est calculée en remplaçant la charge moyenne par son expression obtenue à l'étape 2 dans l'équation décrivant la conservation en flux (2.16).

Remarque 2.2 *La discrétisation ci-dessus est faite dans le cas 2D. Pour l'avoir en 3D, il suffit de remplacer les termes côté par face et aire par volume.*

En ce qui concerne le schéma **en temps**, il est implicite sauf pour le terme de décroissance où on utilise un θ -schéma. Pour plus de précisions, on renvoie à Hoteit & Ackerer [62] et aux thèses Andra de Vincent Martin [76] et Estelle Marchand [73] pour les formulations mathématiques et numériques des équations d'écoulements.

L'utilisation des éléments finis mixtes hybrides permet

- ▷ une définition *a priori* du champ de vitesse à l'aide des fonctions de base de l'espace de Raviart-Thomas,
- ▷ un calcul simultané de la charge et du champ de vitesse,
- ▷ d'assurer la continuité des flux à travers les facettes,
- ▷ l'utilisation de maillage non structuré (2D ou 3D).

La méthode des éléments finis mixtes hybrides donne, dans le cas de l'hydrodynamique, un calcul précis des champs de vitesse. Cependant, à maillage égal, son inconvénient est un nombre d'inconnues plus grand (une par facette).

2.2.3 Discrétisation de l'équation de transport : le code Traces

L'opérateur de transport est la somme de plusieurs opérateurs : la diffusion-dispersion, l'advection, la décroissance radioactive. La technique de séparation d'opérateurs permet de ramener le problème de transport à la résolution d'une suite de problèmes correspondant à

chacun des opérateurs. Ceci permet en particulier de discrétiser différemment l'advection et la diffusion-dispersion et d'utiliser également des pas de temps différents pour la diffusion-dispersion et pour l'advection de sorte que ces pas de temps correspondent à l'échelle de temps de ces différents phénomènes.

L'équation (2.4) est linéarisée par la méthode du point fixe de Picard ([19]) et coupée en deux (splitting) comme ci-dessous :

$$\left\{ \begin{array}{l} \omega \left(\frac{R_{adv}^{n+1,k} C_{adv}^{n+1,k+1} - R_{adv}^{n,k} C_{adv}^{n,k}}{\Delta t} \right) = A(t, C^{n,k}) \quad (2.19) \\ C_{dis}^{n,k} = C_{adv}^{n+1,k+1} \quad (2.20) \\ \omega \left(\frac{R^{n+1,k} C^{n+1,k+1} - R_{dis}^{n,k} C_{dis}^{n,k}}{\Delta t} \right) = B(t, C^{n+1,k+1}) \quad (2.21) \end{array} \right.$$

où k est l'indice d'itération, $R^n = R^n(C^n)$ (dans le cas où le coefficient de retard n'est pas constant), $A(t, C) = -\nabla \cdot (qC)$, $B(t, C) = \nabla \cdot (D\nabla C) - \lambda\omega RC$ et adv désigne l'advection et dis désigne la dispersion. Le critère d'arrêt est donné par

$$\left\| C^{n+1,k+1} - C^{n+1,k} \right\|_2 < \varepsilon_a + \varepsilon_r \left\| C^{n+1,k+1} \right\|_2 \quad (2.22)$$

où ε_a ($\approx 10^{-16}$) est l'erreur absolue et ε_r ($\approx 10^{-12}$) l'erreur relative.

Dans l'équation (2.4), nous avons séparé le flux de concentration en sa partie diffusive $A(t, C)$ et en sa partie convective $B(t, C)$. Cette séparation permet d'utiliser une méthode d'éléments finis discontinus de Galerkin ou une méthode EFMH pour $A(t, C)$ alors que $B(t, C)$ sera obtenu par une méthode EFMH (on construit un système d'équations ayant pour principales inconnues la concentration moyenne par arête). En général, on utilise un schéma implicite pour la partie diffusive et le terme de décroissance radioactive, et un schéma explicite pour la partie convective, avec éventuellement des pas de temps différents pour ses deux parties (plus petits pour la partie convective).

Remarque 2.3 *D'autres méthodes de linéarisation existent, par exemple la méthode de Newton, l'avantage de la méthode Picard réside dans sa simplicité.*

Remarque 2.4 *La séparation d'opérateurs est également utile pour les problèmes où la convection est importante.*

Le code de simulation Traces [62] (Transport RéActif de Contaminant dans les Eaux Souterraines) que j'utilise a été développé par l'Institut de Mécanique des Fluides et des Solides (IMFS) à Strasbourg et par l'Andra. Ce programme, écrit en Fortran 90, simule le flux et le transport en milieu poreux saturé. Le code Traces a été intégré dans la plate-forme de calcul Alliances développée par l'Andra en partenariat avec le CEA et EDF. Traces est basé sur le schéma numérique (2.19)-(2.20)-(2.21).

La version de Traces réalisée à l'IMFS utilisait un schéma implicite pour la partie diffusive et le terme de décroissance, et un schéma explicite pour la partie convective, avec éventuellement des pas de temps différents pour ces deux parties (plus petits pour la partie convective). L'Andra a ajouté un schéma implicite basé sur les éléments finis mixtes hybrides pour la partie convective. La formulation implicite pour la convection nous donne des temps de calculs raisonnables.

Schémas numériques

On rappelle ici les grandes lignes de la méthode des EFDG pour la partie hyperbolique

$$\frac{\partial C}{\partial t} = -\nabla \cdot (q C) \quad (2.23)$$

de l'équation (2.4).

La concentration C est approchée par un espace de fonctions bilinéaires discontinues φ_i^E

$$C_E(x, t) = \sum_{i=1}^{nn} C_E^i \varphi_E^i \quad (2.24)$$

où nn est le nombre de nœuds de l'élément E et C_E^i la valeur de C au nœud i dans E .

L'équation hyperbolique (2.23) est écrite en formulation variationnelle

$$\int_E \frac{\partial C}{\partial t} \varphi_E^i dx = - \int_E \nabla \cdot (q C) \varphi_E^i dx. \quad (2.25)$$

On note par C_A^{in} la variation linéaire de la concentration sur le côté A ($A \subset \partial E$, avec ∂E la frontière de E) à l'intérieur de l'élément E et par C_A^{out} la variation linéaire de C sur A à l'extérieur de E . La discrétisation temporelle est d'ordre deux et est basée sur un schéma explicite simplifié de Runge-Kutta. Pour chaque élément E , les principales étapes de résolution sont

Etape 1 : calcul entre $[t, t + \frac{\Delta t}{2}]$

$$\int_E \frac{C_E^{n+1/2} - C_E^n}{\Delta t/2} \varphi^E dx = \int_E q C_E^n \cdot \nabla \varphi^E dx - \sum_{A \subset \partial E} Q_{E,A} \int_A \frac{C_A^{\text{in},n}}{|A|} \varphi^E d\sigma \quad (2.26)$$

avec φ une fonction test sur E , $Q_{E,A}$ le flux à travers le côté A par rapport à sa normale extérieure $n_{E,A}$ ($\text{sign}(Q_{E,A}) = \text{sign}(q \cdot n_{E,A})$) et $|A|$ est la longueur du côté A .

Etape 2 : calcul de \tilde{C}_E^{n+1} en utilisant les valeurs amont

$$\int_E \frac{\tilde{C}_E^{n+1} - C_E^n}{\Delta t} \varphi^E dx = \int_E q C_E^{n+1/2} \cdot \nabla \varphi^E dx - \sum_{A \subset \partial E} Q_{E,A} \int_A \frac{C_A^{\text{in ou out},n+1/2}}{|A|} \varphi^E d\sigma \quad (2.27)$$

Les flux convectifs sont calculés avec les valeurs amonts de C sur les côtés de A . Le choix entre C_A^{in} et C_A^{out} dépend du signe de $Q_{E,A}$, avec C_A^{in} pour un flux sortant.

Etape 3 : calcul de C_E^{n+1} par stabilisation de la solution avec des limiteurs de pentes

Le schéma numérique est stabilisé avec des limiteurs de pentes multidimensionnels afin d'éviter les oscillations. Pour assurer l'unicité de la solution, on impose à C_E^{n+1} d'être le plus proche possible de \tilde{C}_E^{n+1} . Ainsi, afin de déterminer C_E^{n+1} , on résout alors un problème de minimisation à nn dimensions en utilisant la méthode du point selle.

La solution obtenue vérifie le principe du maximum discret. La résolution en temps étant explicite, le respect de la CFL est obligatoire.

Remarque 2.5 *Tout comme pour les éléments finis mixtes hybrides, la masse est conservée localement.*

2.3 Plateforme de calcul Alliances

2.3.1 Définition & Historique

Une plate-forme de simulation est un logiciel rassemblant tous les outils nécessaires à une simulation et capable d'effectuer le pré-traitement, les calculs et le post-traitement. Le pré-traitement fait référence aux outils nécessaires à la conception d'un problème : la géométrie et le maillage. La géométrie est définie via un logiciel de CAO (Conception Assistée par Ordinateur) tandis que le maillage est défini via un mailleur. Le post-traitement fait référence aux outils nécessaires à la visualisation et à l'analyse des résultats de la simulation. Alliances (Atelier Logiciel d'Intégration, d'ANalyse et de Conception pour l'Entreposage et le Stockage) a été créé afin de permettre l'étude du stockage et de l'entreposage des déchets radioactifs à des échelles de temps importantes. La plate-forme est en constante évolution depuis sa création en 2002 avec l'ajout continue de nouvelles fonctionnalités et de nouveaux modules.

Alliances comprend de nombreux modules qui lui permettent de s'appliquer dans des différents domaines :

- hydraulique et transport ;
- transport réactif et couplage thermique ;
- thermo hydraulique multiphasique ;
- thermo hydro mécanique des argilites ;
- couplage colis environnement ;
- couplage chimie mécanique ;
- analyse d'incertitude et de sensibilité probabiliste.

2.3.2 Fonctionnement de la plate-forme

L'utilisateur cherche à résoudre avec Alliances un problème donné, hydraulique ou transport, avec un code numérique disponible dans la plate-forme. Pour cela, il devra définir (*i.e.* mettre en route) trois objets distincts : un "problème", un "module" et un "composant". La notion de composant est relativement simple : les composants sont les représentants des codes intégrés dans la plate-forme. Informatiquement parlant, ce sont les interfaces python permettant de passer les données d'un problème aux codes de calcul, puis de lancer le calcul. Il y a actuellement trois codes intégrés dans Alliances pour traiter les problèmes cités plus haut :

- * le code **Castem** (Cast3m) du CEA,
- * le code **Porflow** qui est un code commercial vendu par la société Acri,
- * le code **Traces** écrit par l'Institut de Mécanique des Fluides et des Solides (IMFS) de Strasbourg.

Chacun de ces codes est interfacé avec Alliances, via une interface implémentée par code et par problème. Dans une utilisation classique, la mise en route du composant de calcul se fait via les modules. Les modules sont intercalés entre les problèmes et les composants et servent à présenter à l'utilisateur une interface commune pour tous les composants qui implémentent la résolution du problème donné. Dans Alliances, on distingue deux types de problèmes : ceux dont la résolution ne fait intervenir qu'un code de calcul et ceux qui sont couplés au sein d'Alliances et qui font intervenir deux codes de calcul à chaque itération de chaque pas de temps d'un calcul (problèmes dits couplés) comme les problèmes chimie-transport ou hydraulique-transport. Pour ces problèmes couplés, les modules réalisent effectivement le couplage entre les composants (les codes). Tandis que pour les problèmes non couplés, les modules n'agissent qu'en transférant les données depuis

le problème vers le composant choisi par l'utilisateur.

Le modèle de données d'Alliances est constitué des éléments (entités) suivants :

- * Maillages et Champs qui regroupent les méthodes et données d'importation et d'échange de maillage et de champs entre modules. Les maillages, les zones (appelées supports) définies sur ces maillages ainsi que les champs de valeurs (champs de charge, pression ou concentration) s'appuient sur la librairie **Med** de **Salomé**. Alliances traite aussi bien les maillages structurés que non-structurés. On peut également faire des opérations simples sur les champs.
- * Points, Vecteurs, Tenseurs et Tables, objets d'échange pour les données indépendantes d'un maillage. Les Tables désignent un ensemble de colonnes de valeurs, et sont très utilisées pour définir par exemple des données physiques, sources, et/ou conditions aux limites variables dans le temps. Les sorties de type flux ou évolutions en un point sont récupérées sous forme de tables dans un problème instationnaire.
- * Données physiques : on se limitera à celles relatives aux études hydrauliques et de transport généralisé, elles sont regroupées en deux classes : les lois physiques et les quantités physiques (scalaires, vecteurs, tenseurs),
- * Méthodes et données de définition d'un problème physique : type de problème, conditions initiales et aux limites, matériaux, sources, sorties attendues, etc.

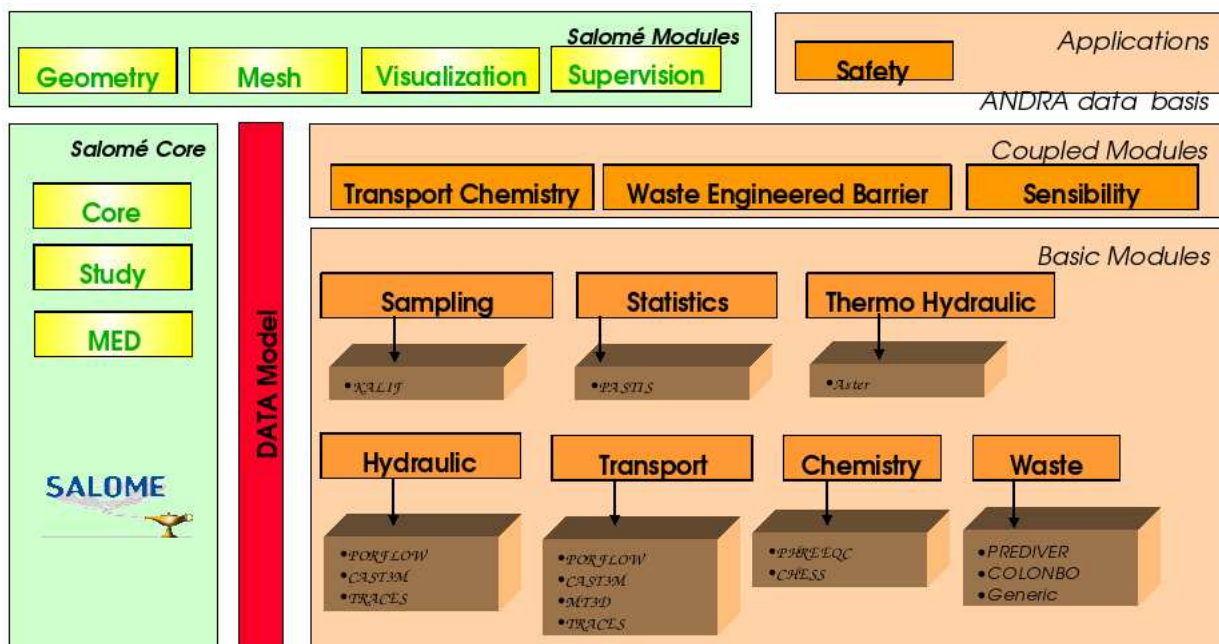


FIG. 2.1 – Architecture d'Alliances

2.3.3 Maillages & Post-traitement

Les maillages ont été générés avec le logiciel Gmshwrapper [50] mis au point par Pierre-Antoine Faure lors d'un stage de Master sous la direction d'Alain Dimier au sein de l'Andra. Ce logiciel encapsule en python le logiciel Open Source Gmsh. Il permet de générer un fichier de maillage au format **Med**, format créé par EDF à partir d'une base HDF*.

Pour le post-traitement, nous avons utilisé l'interface **Salomé** présente dans la plate-

*HDF, Hierarchical Data Format : format développé par le NCSA (National Center for Supercomputing Applications) pour aider les scientifiques à travailler et échanger des données graphiques ou numériques.

forme Alliances et également le logiciel scientifique **Matlab** par l'intermédiaire de scripts que nous avons créés.

2.4 Implémentation

Définition 2.1 *On désigne par variables d'entrées du code les arguments du programme informatique dont la valeur est connue avant l'exécution de cette routine.*

Définition 2.2 *On désigne par variables de sorties du code les arguments du programme informatique dont la valeur est modifiée après l'exécution de celle-ci.*

Définition 2.3 *On appelle routine un morceau de programme composé de fonctions (ensemble d'instructions réalisant une certaine tâche).*

2.4.1 Le code Traces

Traces a été intégré à la version 1.2 de la plate-forme de calcul en novembre 2004. Pour lancer l'exécutable Traces à partir d'un terminal, il faut préciser en entrée deux arguments : le nom du fichier de maillage et le type de calcul demandé (hydraulique, transport, hydraulique + transport, hydraulique transitoire). L'écriture d'un script python qu'on lance au sein de la plate-forme Alliances reste le moyen le plus simple de générer les fichiers d'entrées.

L'architecture de Traces est la suivante (cf. figure ci-dessous) : le programme principal se nomme **traces**, il appelle la routine **traces0** qui, en fonction du type de calcul demandé, appelle soit la routine **comp_hydro** pour résoudre l'hydraulique soit la routine **comp_trans** qui elle-même appelle la routine **comp_specie** afin de résoudre le problème de transport pour chaque espèce.

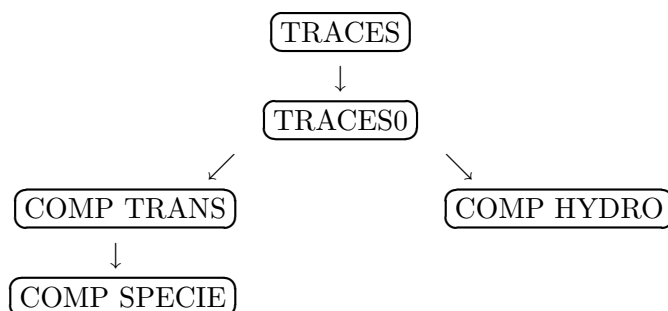


FIG. 2.2 – Architecture de Traces

La version de **traces0** fournie par l'Andra faisait la boucle en temps et procédait à l'écriture des sorties dans des fichiers. Afin de faciliter la différentiation automatique du code Traces (cf. chapitre 4) et de limiter les sources d'erreurs, nous avons séparé la routine principale **traces0** en plusieurs routines : **traces0** ne fait plus que la boucle en temps (partie calculatoire), **onetimestep** fait le calcul sur un pas de temps et **calcul_traces** fait la partie lecture/écriture de fichiers. Nous avons rajouté la routine **fct_cout** qui calcule la fonction coût et nous désignons par **cost** la valeur de la fonction coût (variable de sortie du code).

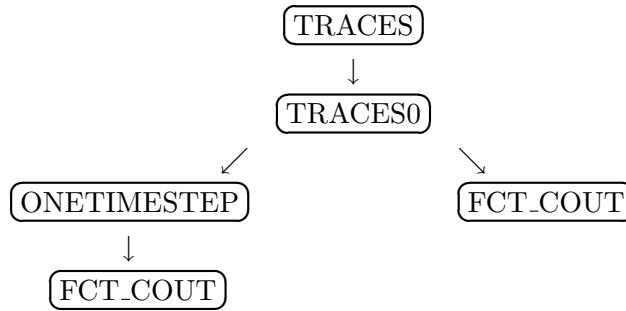


FIG. 2.3 – Structure du fichier traces0.f90

D'autres fichiers que nous ne présentons pas ici ont également été modifiés afin d'avoir une version de Traces plus adaptée pour la partie différentiation automatique du code (chapitre 4).

Remarque 2.6 La routine *fct_cout* est appelée deux fois dans *traces0* car la fonction coût (3.2) associée au problème inverse (3.1) est la somme de deux termes : un terme correspondant à l'intégration en temps et un terme correspondant à l'instant final.

2.4.2 La plate-forme Alliances

En général, les différents arguments constituant un problème sont définis au préalable, pour plus de lisibilité du script notamment. Un ordre raisonnable de définition de ces données est le suivant :

1. définition des espèces (en transport) ;
2. définition des données sur les espèces (en transport) : la décroissance/filiation ;
3. définition des paramètres matériaux (intrinsèques et matériaux/espèces) ;
4. création des matériaux ;
5. chargement du maillage ;
6. récupération des zones et des frontières ;
7. création des régions (association zones/matériaux) ;
8. spécification du type de problème ;
9. définition des conditions initiales ;
10. définition des conditions aux limites ;
11. définition des instants de calcul ;
12. définition des sources (s'il y a lieu) ;
13. définition des sorties attendues (concentration, flux, etc.) ;
14. création du problème d'hydraulique en saturé ou du problème de transport généralisé en passant l'ensemble des données en argument.

2.5 Position du problème

Nous allons maintenant préciser les caractéristiques physiques et la géométrie des domaines considérés afin de formuler un modèle complet pour la simulation du transport

de radionucléides. Le modèle choisi par l'Andra pour le dossier 2005 ([8]) a validé l'approximation du milieu et du stockage par des couches horizontales, on se limitera donc à une description bidimensionnelle du problème. L'objectif étant de caractériser un comportement global et de valider une approche théorique sur ce comportement, la géométrie envisagée est simplifiée en négligeant certains détails de l'installation (galeries, alvéoles, puits, etc.) et du milieu géologique.

2.5.1 Cadres et objectifs de notre travail

Comme nous l'avons expliqué dans le chapitre 1, l'Andra étudie la possibilité de stocker les déchets nucléaires de haute activité (HA) et de moyenne activité à vie longue (MA-VL) en couche géologique profonde. Le laboratoire de Bure dans la Meuse/Haute Marne, dont la configuration du sous-sol est donnée par la figure 2.4, a été créé dans ce but. La couche argileuse du Callovo-Oxfordien (Cox), située à 500 m de profondeur et mesurant 130 m d'épaisseur, est susceptible d'accueillir le stockage.

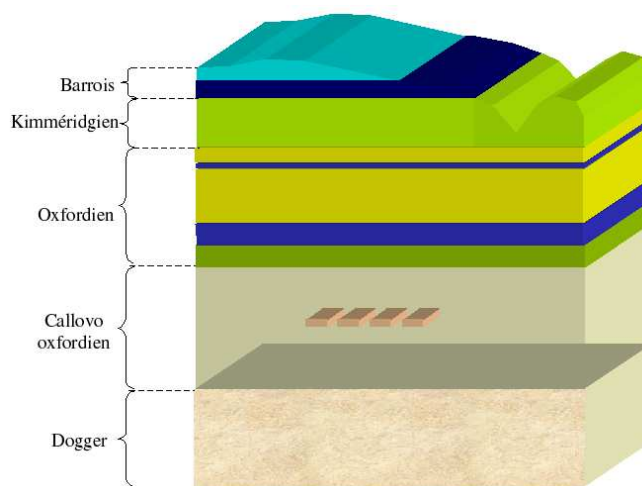


FIG. 2.4 – Représentation du milieu géologique

Afin d'étudier l'évolution d'un stockage et de son environnement sur des échelles de temps hors du champ de l'expérience (jusqu'à plusieurs centaines de milliers d'années), il convient d'utiliser des outils adaptés tels que la simulation numérique. L'évaluation de la sûreté du stockage s'appuie notamment sur la simulation des écoulements et du transport de radionucléides dans le milieu géologique. L'objectif de ce travail est de pouvoir appliquer les résultats obtenus dans cette thèse à l'expérience DIR. Dans ce but, nous avons défini des cas-tests en liaison avec les équipes de l'Andra afin de valider les méthodes utilisées. Dans cette optique, nous avons mis au point avec les équipes de l'Andra, deux cas-tests 2D correspondant à des situations physiques réalistes que nous présentons dans le paragraphe suivant.

2.5.2 Présentation des cas-tests

Le radionucléide considéré est l'iode 129 dont la période radioactive est de 15,7 millions d'années et le coefficient de décroissance radioactive vaut $\lambda = \log(2)/1,57 \cdot 10^7 \text{ an}^{-1}$. La gamme de valeur des paramètres associée à l'iode pour le Cox est présentée dans le tableau ci-dessous.

	D (m^2/s)	α_L (m)	α_T (m)	ω	R
Callovo-Oxfordien	5.10^{-12}	7	0,7	0,05	1

TAB. 2.1 – Coefficients pour l’Iode 129

Les paramètres de porosité et de diffusion sont définis par zones où ils sont considérés comme constants, on dit que la variabilité spatiale des paramètres du système est paramétrée par zones.

La géométrie du premier cas-test, appelé cas-test “colis”, est représentée sur la figure 2.5 : il modélise un colis contenant des radioéléments (Ω_2) placé dans une couche géologique (Ω_1). Le second, appelé cas-test “réel” (car proche de l’expérience DIR), peut modéliser le sous-sol du laboratoire de Bure : le trou représentant le forage (cf. figure 2.6), le domaine Ω_2 le colis, le domaine Ω_3 la zone endommagée par le forage et la zone Ω_1 la roche. Les valeurs des paramètres sont données par le tableau 2.2 : partie gauche pour le cas-test “colis” et partie droite pour le cas-test “réel”.

	D (m^2/an)	ω	R		D (m^2/s)	ω	R
Ω_1	$9,5.10^{-7}$	0,3	1	Ω_1	10^{-11}	0,2	1
Ω_2	2.10^{-4}	10^{-3}	1	Ω_2	10^{-6}	10	1
				Ω_3	10^{-10}	0,3	1

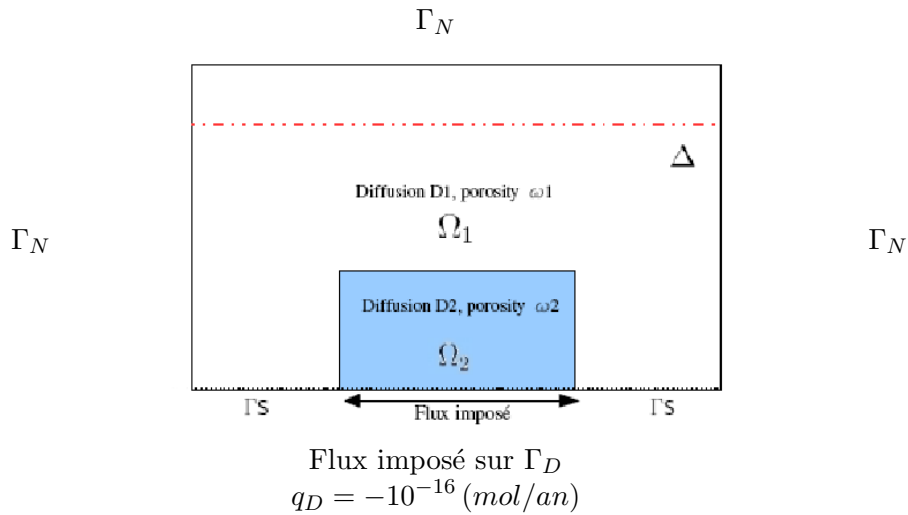
TAB. 2.2 – Coefficients de diffusion/dispersion, de porosité et de retard

La valeur de la porosité dans Ω_2 (zone d’injection) pour le cas-test “réel” a volontairement été fixée à une valeur élevée afin de rendre disponible très rapidement les radio-nucléides à la frontière de la zone endommagée.

2.5.2.1 Cas-test “colis”

Le domaine Ω est découpé en deux zones et mesure 60 m de longueur par 40 m de largeur. La zone Ω_2 mesure 20 m de longueur par 10 m de largeur. Les valeurs de diffusion ont été choisies proches de la valeur réelle du Callovo-Oxfordien pour Ω_1 et proche des valeurs des couches géologiques voisines du Callovo-Oxfordien pour Ω_2 (cf. tableau 2.1). On impose un flux (non nul) sur la frontière Γ_D et une concentration initiale nulle dans Ω_2 .

Dans la suite, on calculera les valeurs de concentration dans le colis à tous les pas de temps et sur la droite Δ (en pointillée rouge sur la figure 2.5) à l’instant final pour calculer la fonction coût du problème inverse. Les valeurs de sorties du code seront par la suite considérées comme des mesures (données) pour l’identification.

FIG. 2.5 – Représentation du domaine Ω (60 m \times 40 m)

2.5.2.2 Cas-test “réel”

La figure 2.6 donne la géométrie du domaine du cas-test “réel”. Le choix de la géométrie est totalement en accord avec la figure 2.7 qui montre une partie du dispositif expérimental (la chambre de diffusion) de l’expérience DIR. Le domaine Ω est divisé en trois zones et mesure 0,4 m de longueur par 2 m de largeur, la zone Ω_2 mesurant 8 cm de longueur par 10 cm de largeur et Ω_3 ayant une largeur de 2 cm. Les droites $\gamma_i, i = 1, \dots, 3$ (en pointillées rouges sur la figure 2.6) représentent les endroits où la concentration sera mesurée à l’instant final pour calculer la fonction coût de notre problème inverse. Les sorties du code seront (comme pour le cas-test “colis”) considérées comme des mesures pour le problème d’identification.

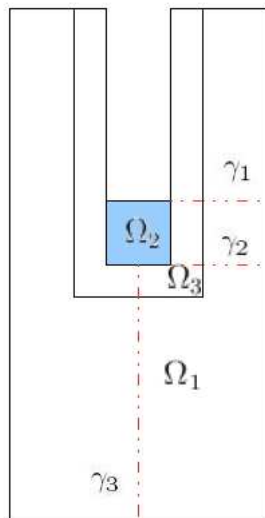
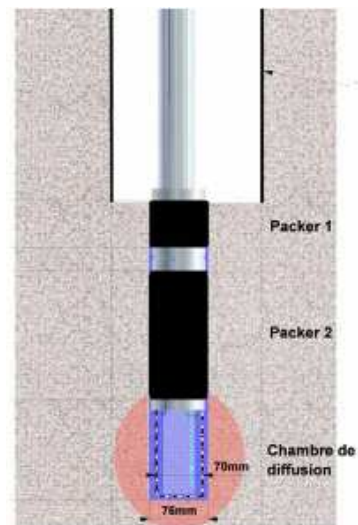
FIG. 2.6 – Représentation du domaine Ω (0,4 m \times 2 m)

FIG. 2.7 – Chambre de diffusion

Nous avons traité seulement le cas 2D alors que la géométrie du vrai problème est en 3D (axisymétrique). Cependant, cela ne remet pas en cause la démarche adoptée mais cela nous a permis de gagner du temps calcul.

2.6 Résolution du problème

Dans les applications qui suivent, la vitesse de Darcy est nulle. On aura donc à résoudre un problème de diffusion pure (pas de convection). La géométrie et les dimensions des deux cas-tests sont définis à la section 2.5.2.

2.6.1 Cas-test “colis”

On note la frontière $\partial\Omega$ de Ω et on a $\partial\Omega = \Gamma_N \cup \Gamma_S \cup \Gamma_D$ (cf figure 2.5). On impose un flux de -10^{-16} mol/an sur la frontière Γ_D et le terme source Q de l'équation (1.6) est nul. Les conditions initiales et aux limites sont les suivantes

$$\left\{ \begin{array}{l} C(t=0, x) = 0 \text{ sur } \Omega_1 \\ C(t=0, x) = 0 \text{ sur } \Omega_2 \end{array} \right. \quad (2.28) \quad \left\{ \begin{array}{l} C(t, x) = 0 \text{ sur } \Gamma_N \\ D\nabla C(t, x).n = 0 \text{ sur } \Gamma_S \\ D\nabla C(t, x).n = -10^{-16} \text{ sur } \Gamma_D \end{array} \right. \quad (2.29)$$

Le domaine Ω a été maillé avec gmsh, il contient 2742 éléments (figure 2.8) dont 2500 triangles ou quadrangles pour la grande zone (Ω_1) et 242 rectangles pour le colis (Ω_2).

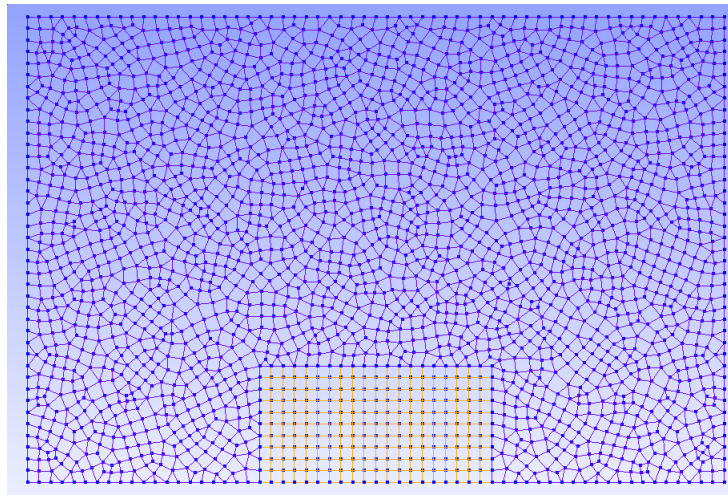


FIG. 2.8 – Domaine Ω maillé avec 2742 éléments

La simulation s'est déroulée sur 1000 ans avec des pas de temps d'une demi-année. On peut voir le résultat de la simulation sur la figure 2.9. Les coefficients de diffusion étant assez faibles, la diffusion dans la zone du colis n'est pas très importante : on voit en effet que la concentration n'a pas complètement atteint le haut du colis.

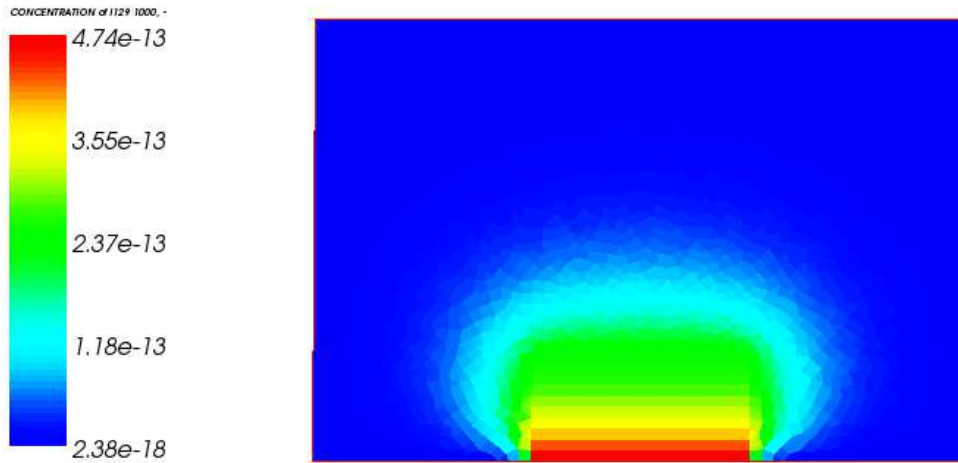


FIG. 2.9 – Visualisation de la concentration obtenue avec Salome à l'instant final

2.6.2 Cas-test “réel”

On définit $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ et $\Gamma = \partial\Omega$. On prend comme conditions initiales et aux limites

$$\begin{cases} C(t=0, x) = 0 & \text{sur } \Omega_1 \cup \Omega_3 \\ C(t=0, x) = 1 & \text{sur } \Omega_2 \end{cases} \quad (2.30) \quad D\nabla C(t, x) \cdot n = 0 \text{ sur } \Gamma \quad (2.31)$$

Le problème étant symétrique, nous le résolvons uniquement sur la moitié du domaine qui a été maillé avec 1996 éléments. On prend le terme source Q nul dans l'équation (1.6) et on part d'une concentration de 1 mol/m^3 dans la zone d'injection Ω_2 qu'on laisse évoluer sur 1000 jours (pas de temps d'une demi-journée). On peut voir le résultat sur la figure 2.10 : la concentration a légèrement diminué dans le colis mais elle a cependant atteint le bord droit du domaine.

Pas de temps 2001

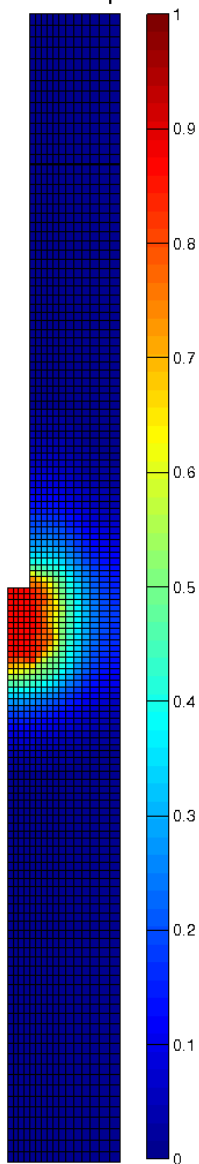


FIG. 2.10 – Visualisation de la concentration obtenue avec Matlab à l'instant final

Les profils 3D de concentration (figures 2.11, 2.12 et 2.13) représentent l'évolution au cours du temps de la concentration sur les trois droites $\gamma_i, i = 1, \dots, 3$: sur l'axe des z nous avons les valeurs de concentration, sur l'axe des abscisses nous avons le temps et sur l'axe des ordonnées nous avons la première composante d'espace x pour les figures 2.11 et 2.12 et la deuxième composante d'espace y pour la figure 2.13. Le temps est représenté à chaque pas de temps d'une demi-journée, la concentration évolue donc de $t = 0$ à $t = 1000$ jours ce qui correspond à la numérotation de 0 à 2000 sur les figures. On a également représenté la solution sur tous les pas de temps d'espace (de longueur 0,2 cm) du maillage, la composante x pouvant aller de 0 à 0,2 m et la composante y de 0 à 2 m (cf. section 2.5.2.2). Les profils horizontaux (cf. figure 2.6) vont du bord droit du colis au bord droit du domaine (de 0,04 à 0,2 m) et le profil vertical (cf. figure 2.6) va du bord du colis au bas du domaine (de 1,1 à 0 m).

Sur les figures, on voit bien la décroissance exponentielle de la diffusion qui est plus ou moins forte suivant les zones (les changements de pente correspondent aux changements de zones). Ces résultats sont bien en accord avec la physique et avec ce qu'on pouvait attendre.

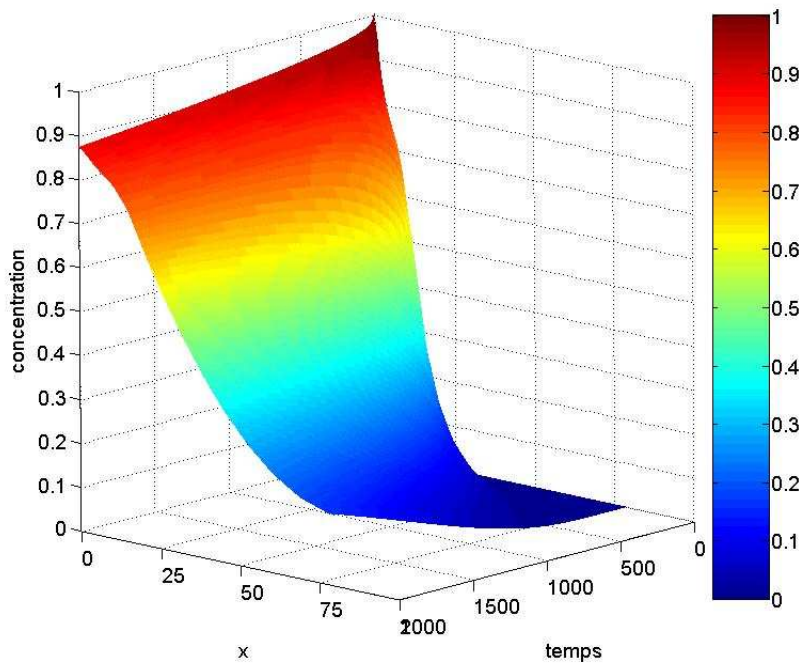


FIG. 2.11 – Profil de concentration sur la ligne horizontale γ_1 d'équation $y = 1$

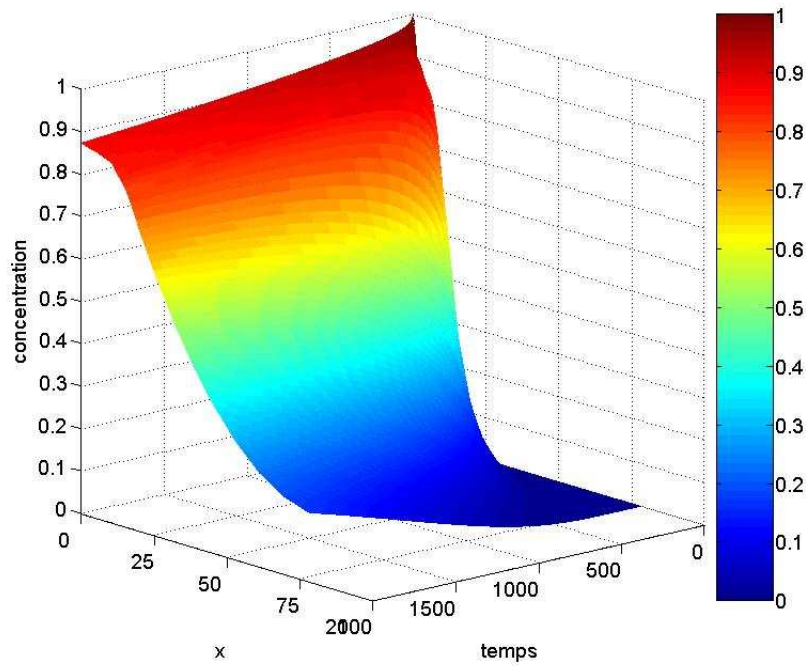


FIG. 2.12 – Profil de concentration sur la ligne horizontale γ_2 d'équation $y = 0,9$

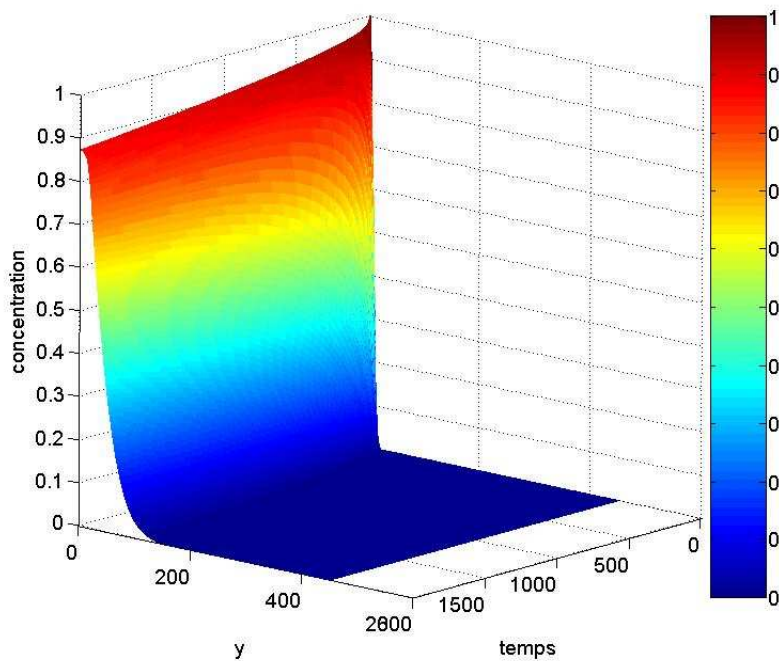


FIG. 2.13 – Profil de concentration sur la ligne verticale γ_3 d'équation $x = 0$

Chapitre 3

Etude du problème inverse

Sommaire

3.1	Introduction	36
3.1.1	Etat de l'art	36
3.1.2	Cadres et objectifs de notre travail	37
3.1.2.1	Les objectifs	37
3.1.2.2	Les données à notre disposition	37
3.1.2.3	Identifiabilité	37
3.2	Etude théorique du problème inverse	38
3.2.1	Rappels de contrôle optimal	38
3.2.2	Le modèle linéaire tangent	39
3.2.3	Introduction de l'état adjoint	40
3.2.3.1	Approche lagrangienne	40
3.2.3.2	Approche "linéarisée"	43
3.2.3.3	Régularisation	43
3.2.4	Obtention formelle de l'équation de transport adjointe	43
3.2.4.1	Cas-test "colis"	43
3.2.4.2	Cas-test "réel"	47
3.3	Etude numérique du problème inverse	52
3.3.1	Introduction	52
3.3.2	Le code adjoint	52
3.3.3	Rappels sur les méthodes d'optimisation	53
3.3.4	Description de l'algorithme de minimisation	54
3.3.5	Implémentation	55

Avant de présenter les méthodes de résolution des problèmes inverses, il convient de définir ce qu'on appelle par problème inverse : on dit résoudre un **problème inverse** (par opposition au **problème direct**, cf. chapitre 2) lorsqu'on cherche à déterminer les paramètres de porosité ω et de diffusion D dans l'équation (1.6) connaissant les conditions initiales et aux limites ainsi que les valeurs de concentrations.

Dans ce chapitre, nous rappelons dans un premier temps les différentes méthodes de résolution des problèmes inverses et les bases du contrôle optimal avant d'écrire formellement les équations de transport adjointes. Dans un second temps, nous passons à l'étude numérique du problème inverse.

Dans toute la suite, on note $\omega = (\omega_1, \dots, \omega_p)$, $D = (D_1, \dots, D_p)$ et $K = (\omega, D)$ le vecteur des paramètres. On pose $N = 2p$.

3.1 Introduction

3.1.1 Etat de l'art

Une littérature abondante existe sur la résolution des problèmes inverses, on peut citer l'ouvrage de synthèse [21] qui fait suite à un travail à l'Observatoire Français des Techniques Avancées et en particulier le chapitre «Des apports du contrôle optimal» de Jacques Blum. Un workshop a également été organisé en novembre 2002 pour faire l'état de l'art sur le thème «Problèmes inverses et analyse d'incertitudes» dans le cadre du GNR MoMaS avec les partenaires de l'Andra et du CEA*.

On peut également se référer à Kern [65] pour une approche générale sur les problèmes inverses, aux travaux de Mc Laughlin et Townley [77] pour les méthodes d'inversion, Sun [90] pour des problèmes inverses en milieu poreux et De Marsily et al [45] pour les problèmes inverses en hydrogéologie et Chavent et al. [37] pour une estimation des perméabilités relatives en milieu poreux. Il faut mentionner Cartalade et al [30] dans le cadre de l'estimation des paramètres de transport ou de diffusion en milieu poreux et dans l'approche par réseaux de neurones pour l'identification des coefficients de diffusion [29]. Enfin citons Clément et al [41] pour l'analyse de sensibilité de la solution de l'équation aux dérivées partielles par rapport aux coefficients à estimer à l'aide de l'algorithme de décomposition en valeurs singulières.

Pour résoudre notre problème inverse, on utilisera des méthodes déterministes ou encore méthodes variationnelles basées sur la minimisation d'un critère en général quadratique. Ces méthodes trouvent leur fondement mathématique dans la théorie du contrôle optimal de Lions [70], appliquée à l'estimation de paramètres comme dans Chavent [33]. Il existe aussi des méthodes de type probabilistes pour résoudre un problème inverse comme exposé dans Tarantola [93]. Quelle que soit la méthode de résolution employée, la solution est obtenue en utilisant une approche identique. En effet, un critère est souvent défini pour mesurer la différence entre valeurs calculées et valeurs mesurées : l'objectif est de trouver l'ensemble des paramètres qui minimisent le critère. Il reste à choisir une technique appropriée pour minimiser la fonction coût. De nombreuses approches existent pour faire une telle minimisation ([23, 40, 61] et cf. section 3.3) : nous avons choisi d'utiliser des algorithmes qui requièrent le calcul du gradient (dérivées de la fonction coût par rapport aux paramètres).

Nous pouvons souligner que les méthodes inverses dépendent du type de paramétrisation choisie pour caractériser les paramètres à identifier dans le problème direct. Dans le cas

*http://www.gdrmomas.org/Activites/2002/Documents/WORKSHOP_control.doc

de l'identification de paramètres, plusieurs méthodes peuvent être utilisées : méthode de zonation (où l'on suppose les coefficients constants par zones), méthode d'interpolation (approximation polynomiale), etc. Dans notre cas, la représentation du milieu géologique nous a amené à paramétrer le système par zones. Ce choix est raisonnable puisque le système est naturellement paramétré par zones. La même méthode s'applique dans le cas des strates géologiques en milieu naturel [26] où l'on pourra envisager une zonation horizontale (cf. figure 2.4).

3.1.2 Cadres et objectifs de notre travail

Nous devons faire des simulations du transport des radionucléides afin de déterminer l'impact d'une éventuelle propagation de ces éléments dans le sous-sol. La modélisation de l'écoulement et du transport en milieu poreux autour du stockage exige de connaître les paramètres physiques des différentes couches géologiques. Parmi ces paramètres, la porosité et la diffusion ne sont pas directement accessibles par des mesures, c'est pourquoi nous devons résoudre un problème inverse pour les identifier.

3.1.2.1 Les objectifs

La première partie de notre travail est d'estimer les coefficients de transport à partir de mesures de concentration. La spécificité de l'approche choisie est d'utiliser la technique de zonation qui consiste à considérer le milieu comme un ensemble de zones homogènes, pour chacune desquelles les coefficients seront estimés. Cette approche pourra être appliquée à des expériences dans des milieux dont l'échelle d'espace peut être très différente. Le forage, dans le cas de l'expérience DIR et du cas-test "réel", implique bien sûr une identification des coefficients à la fois dans la zone endommagée et dans la zone saine. Dans un premier temps, on commencera par identifier les coefficients dans des zones fixées. Pour cela, on a été amené à développer le code adjoint du code de diffusion-convection. En effet, le code adjoint nous donnera accès au gradient de la fonction coût dont nous avons besoin dans la résolution du système d'optimalité (cf. paragraphe 3.2.3). Dans un deuxième temps, on se préoccupera de déplacer les zones et éventuellement d'en créer de nouvelles (cf. chapitre 6).

3.1.2.2 Les données à notre disposition

Expériences jumelles

Nous allons travailler sur des données synthétiques : on choisit tout d'abord des valeurs précises pour le coefficient de diffusion et la porosité et on calcule la concentration qui en résulte (noté C_{obs}) en résolvant le problème direct. Ensuite, on cherche à retrouver ces coefficients à partir des valeurs de concentration observées. L'avantage de cette méthode est que l'on peut comparer la porosité et le champ de diffusion obtenus par la méthode inverse avec la porosité et le champ de diffusion initiaux (qui sont en pratique inconnus) ce qui permettra de valider les méthodes utilisées.

Dans les deux cas-tests qui nous intéressent, les données de concentration observées sont de deux types : des mesures de concentration recueillies dans la chambre d'injection tout au long de l'expérience ainsi que des mesures obtenues au temps final.

3.1.2.3 Identifiabilité

La question se pose toujours, de façon théorique et pratique, de savoir si les mesures sont suffisantes pour estimer les paramètres que l'on cherche et comment l'incertitude sur

les mesures se répercute sur les paramètres estimés. Les questions de stabilité et d'identifiabilité sont donc au cœur de l'estimation de paramètres et sont intimement liées au choix de la paramétrisation. Nous avons réalisé de nombreuses simulations dans le chapitre 5 pour essayer d'y répondre.

3.2 Etude théorique du problème inverse

Dans ce chapitre, nous présentons le problème inverse d'identification de paramètres en milieu poreux pour une équation de convection-diffusion. Nous chercherons la solution d'un problème type contrôle optimal connaissant une distribution de la solution en différents points de l'espace et du temps. A cet effet, nous écrirons l'équation adjointe de (1.6). L'état adjoint permettra alors de déterminer le gradient d'une fonction coût définie en fonction des paramètres à identifier. Les observations seront générées par le modèle direct (on parle de données synthétiques) ce qui permettra, connaissant les valeurs des coefficients à identifier, de valider les méthodes utilisées.

Notre problème inverse se formulant comme un problème d'optimisation (cf. section 3.1), nous rappelons ici les grandes lignes de la théorie du contrôle optimal qui nous servira par la suite.

3.2.1 Rappels de contrôle optimal

Principe

Dans la théorie du contrôle optimal, le modèle peut être décrit par une équation d'évolution non linéaire (ce modèle peut être la version discrétisée en espace, par la méthode des différences finies par exemple, d'un modèle d'équation aux dérivées partielles) de la forme

$$\begin{cases} \frac{dC}{dt} = F(C(t), K), & t \in [0, T], C \in \mathcal{H} \\ C(0) = U \end{cases} \quad (3.1)$$

où C est la variable d'état appartenant à un espace de Hilbert \mathcal{H} ($\mathcal{H} \equiv L^2(0, T; \mathbb{R}^m)$), K le vecteur des paramètres appartenant à un espace de Hilbert \mathcal{K} , F un opérateur différentiel non linéaire de $\mathcal{H} \times \mathcal{K}$ dans \mathcal{H} qu'on supposera (en général) différentiable. On supposera que l'ensemble de paramètres admissibles \mathcal{P} est un espace de Hilbert et que $\mathcal{P} = \mathcal{K}$. On désigne par \mathcal{O} l'espace de Hilbert des observations. On ne précise pas plus les espaces fonctionnels associés (ce n'est effectivement pas le but de l'étude). Lors des applications numériques, \mathcal{O} sera nécessairement un espace de dimension finie. Nous supposons que pour $K \in \mathcal{K}$ et $C \in \mathcal{H}$ fixés, le problème de Cauchy (3.1) admet une solution unique $C = C(K)$.

Nous devons tenir compte des mesures qui sont à notre disposition, nous les représenterons par une application linéaire, notée M_1 (appelée opérateur d'observation) de notre fonction d'état C à valeurs dans \mathcal{O} pour la partie temporelle de la fonction coût et par une application linéaire, notée M_2 (appelée opérateur d'observation à l'instant final) de notre fonction d'état C à valeurs dans \mathcal{U} pour le terme portant sur l'instant final. On définit une fonction coût J de \mathcal{P} dans \mathbb{R} (ou critère d'erreur) qui calcule l'écart entre solution calculée par le modèle (3.1) d'une part et une observation "temporelle" $C^o \in \mathcal{O} \times [0, T]$ et une observation au temps final $C_{\text{final}}^b \in \mathcal{U}$ d'autre part

$$J(K) = \alpha \int_0^T \|M_1 C - C^o\|_{\mathcal{O}}^2 dt + \beta \|M_2 C(T) - C_{\text{final}}^b\|_{\mathcal{U}}^2 \quad (3.2)$$

où $\|\cdot\|^2$ est la norme euclidienne usuelle et α, β des coefficients réels généralement appelés poids. La fonction coût J contient

- un terme $\alpha \int_0^T \|M_1 C - C^o\|_{\mathcal{O}}^2 dt$ où les valeurs de concentration sont calculées dans un domaine donné à plusieurs instants,
- un terme $\beta \|M_2 C(T) - C_{\text{final}}^b\|_{\mathcal{U}}^2$ où les valeurs de concentration sont prises à l’instant final.

On pose alors le problème de contrôle optimal suivant :

Problème 3.1 *Trouver $K^* = (\omega^*, D^*) \in \mathcal{P}$ tels que*

$$J(K^*) = \inf_{K \in \mathcal{P}} J(K) \quad (3.3)$$

où J est donné par (3.2) et C est associé à K par (3.1).

Remarque 3.1 *Sans hypothèses sur l’opérateur F et sur l’opérateur d’observation M_1 , l’optimum K^* peut ne pas exister ou ne pas être unique. Pour plus de détails sur la résolution des problèmes de contrôle optimal, on renvoie au livre référence de Lions [70].*

Remarque 3.2 *Nous limitons nos rappels au cas de la dimension finie.*

De façon générale, lorsque J est différentiable, une condition nécessaire pour que K^* soit solution de (3.3) dans le cas sans contraintes est donnée soit par l’équation d’Euler-Lagrange

$$\nabla J(K^*) = 0 \quad (3.4)$$

avec $\nabla J = \left(\frac{\partial J}{\partial \omega}, \frac{\partial J}{\partial D} \right)^T$, et dans le cas avec contraintes par l’inégalité variationnelle

$$\langle \nabla J(K^*), v - K^* \rangle \geq 0, \quad \forall v \in \mathcal{P}. \quad (3.5)$$

La fonction coût J dépend de K par l’intermédiaire de C , c’est la différence avec un problème “classique” de minimisation. La détermination de ∇J permet de mettre en œuvre des méthodes d’optimisation de type descente. La méthode directe consiste à dériver C par rapport à K , ce qui revient à dériver l’équation (1.6) par rapport à chaque composante de K et donc à écrire autant d’équations linéarisées que de paramètres à identifier. Ce procédé devient d’autant plus coûteux que le nombre de paramètres est grand. C’est pourquoi, dans la suite nous allons présenter (et utiliser) la méthode de l’état adjoint qui permet de calculer le gradient en intégrant l’équivalent de deux fois le modèle direct.

3.2.2 Le modèle linéaire tangent

Si on note \tilde{K} une perturbation sur K alors cette perturbation permet de définir la dérivée directionnelle \hat{J} de la fonction coût J dans la direction K

$$\hat{J}(K; \tilde{K}) = 2\alpha \int_0^T \langle M_1 C - C^o, M_1 \hat{C} \rangle dt + 2\beta \langle M_2 C(T) - C_{\text{final}}^b, M_2 \hat{C}(T) \rangle \quad (3.6)$$

où \widehat{C} désigne la dérivée de Gâteaux du modèle direct (3.1). Alors \widehat{C} est solution de l'équation (3.7) appelée modèle linéaire tangent

$$\begin{cases} \frac{d\widehat{C}}{dt} = \left(\frac{\partial F}{\partial C}\right) \widehat{C} + \left(\frac{\partial F}{\partial K}\right) \widetilde{K}, & t \in [0, T], \widehat{C} \in \mathcal{H} \\ \widehat{C}(0) = 0 \end{cases} \quad (3.7)$$

où $\left(\frac{\partial F}{\partial C}\right)$ et $\left(\frac{\partial F}{\partial K}\right)$ représentent respectivement l'opérateur Jacobien de F par rapport à C et par rapport à K .

Par définition du gradient, on a

$$\widehat{J}(K, \widetilde{K}) = \langle \nabla J(K), \widetilde{K} \rangle. \quad (3.8)$$

Une façon de déterminer le gradient (c'est-à-dire toutes ses composantes) est de faire une perturbation unitaire sur chaque paramètre, en maintenant les autres constants, et de calculer \widehat{J} . Ceci implique qu'il faut résoudre (3.7) autant de fois qu'il y a de paramètres. Dans notre cas, il y a plusieurs paramètres (autant que de zones) à identifier et il faut faire appel au gradient dans un algorithme de minimisation. On comprend donc très vite que le coût (calcul et espace mémoire) de cette démarche devient vite élevé. On introduit donc dans la suite une méthode de calcul du gradient qui **nécessite seulement une intégration** du modèle direct **et une intégration** d'une équation associée au modèle linéaire tangent, indépendamment du nombre de paramètres de contrôle.

3.2.3 Introduction de l'état adjoint

La méthode de l'état adjoint consiste à voir l'équation d'état (3.1) comme une contrainte d'égalité reliant la variable d'état C (la concentration) aux paramètres ω et D (variables de contrôle). C est donc supposé a priori indépendant de $K = (\omega, D)$.

Le problème inverse consiste à retrouver le paramètre K qui reproduirait au mieux les mesures C^o et C_{final}^b : ceci correspond au problème de minimisation (3.3). Nous résolvons ce problème de minimisation par un algorithme de descente (cf. section 3.3), nécessitant le calcul du gradient de la fonctionnelle coût (3.2) que nous allons obtenir par la méthode de l'état adjoint.

3.2.3.1 Approche lagrangienne

Nous considérons alors le **multiplicateur de Lagrange** P associé à l'équation (3.1), qui est ici une fonction test (dépendant du temps et de l'espace) de \mathcal{H} . Notons qu'à ce stade, la valeur de la variable adjointe P n'est pas encore fixée. Nous formons le **Lagrangien** du problème en additionnant la fonction coût (3.2) et le produit scalaire de P par l'équation (3.1) qu'on intègre sur $[0, T]$. Nous obtenons l'expression

$$\begin{aligned} \mathcal{L} = \mathcal{L}(C, K; P) &= \alpha \int_0^T \|M_1 C - C^o\|_{\mathcal{O}}^2 dt + \beta \|M_2 C(T) - C_{\text{final}}^b\|_{\mathcal{U}}^2 \\ &\quad + \int_0^T \left\langle \frac{dC}{dt} - F(C, K), P \right\rangle dt. \end{aligned} \quad (3.9)$$

Ce Lagrangien est une fonction à valeurs dans \mathbb{R} que nous supposons être Gâteaux-dérivable par rapport aux trois variables C , K et P . On rappelle la définition de Gâteaux-dérivée :

Définition 3.1 (*Gâteaux-dérivée ou dérivée directionnelle*) Soit φ une application de \mathbb{R}^N dans \mathbb{R} de variable v . On dit que φ est G -dérivable dans la direction w , si la limite

$$\widehat{\varphi}(v, w) = \lim_{\theta \rightarrow 0} \frac{\varphi(v + \theta w) - \varphi(v)}{\theta}$$

existe et si l'application $\varphi'(v) : w \mapsto \widehat{\varphi}(v, w)$ est linéaire continue de \mathbb{R}^N dans \mathbb{R} . Si c'est le cas, alors on pourra noter

$$\widehat{\varphi}(v, w) = \langle \nabla \varphi(v), w \rangle$$

où $\nabla \varphi(v)$ est le gradient de φ par rapport à v .

Quand $C(K)$ est solution de l'équation d'état (3.1), nous avons

$$J(K) = \mathcal{L}(C(K), K; P). \quad (3.10)$$

Nous supposons que l'application $K \mapsto C(K)$ est différentiable (cf. Lions [70]) et on poursuit notre résolution formelle du problème (3.3) en calculant alors la Gâteaux-dérivée par rapport à K dans les deux membres de l'équation précédente. Nous arrivons à la relation

$$\widehat{J} = \frac{\partial \mathcal{L}}{\partial C}(C(K), K; P) \widehat{C} + \frac{\partial \mathcal{L}}{\partial K}(C(K), K; P) \widetilde{K} \text{ pour tout } P. \quad (3.11)$$

Nous choisissons le multiplicateur de Lagrange P de telle sorte que le premier terme de l'équation précédente soit nul, *i.e.*

$$\frac{\partial \mathcal{L}}{\partial C}(C(K), K; P) = 0. \quad (3.12)$$

Cette égalité se réécrit (après intégration par parties) de la façon suivante

$$\begin{aligned} & 2\alpha \int_0^T \langle M_1 C - C^o, M_1 \widetilde{C} \rangle dt + 2\beta \langle M_2 C(T) - C_{\text{final}}^b, M_2 \widetilde{C}(T) \rangle - \int_0^T \left\langle \frac{dP}{dt}, \widetilde{C} \right\rangle dt \\ & + \langle P(T), \widetilde{C}(T) \rangle - \int_0^T \left\langle \left(\frac{\partial F}{\partial C} \right) \widetilde{C}, P \right\rangle dt = 0. \end{aligned} \quad (3.13)$$

Comme nous sommes dans un espace de Hilbert, nous pouvons introduire les opérateurs adjoints de M_1^* et $\left(\frac{\partial E}{\partial C} \right)^*$ dans (3.13)

$$\begin{aligned} & 2\alpha \int_0^T \langle M_1^*(M_1 C - C^o), \widetilde{C} \rangle dt + 2\beta \langle M_2^*(M_2 C(T) - C_{\text{final}}^b), \widetilde{C}(T) \rangle - \int_0^T \left\langle \frac{dP}{dt}, \widetilde{C} \right\rangle dt \\ & + \langle P(T), \widetilde{C}(T) \rangle - \int_0^T \left\langle \left(\frac{\partial F}{\partial C} \right)^* P, \widetilde{C} \right\rangle dt = 0. \end{aligned} \quad (3.14)$$

Soit encore

$$\begin{cases} -\frac{dP}{dt} & = \left(\frac{\partial F}{\partial C} \right)^* P - 2\alpha M_1^*(M_1 C - C^o) \\ -P(T) & = 2\beta M_2^*(M_2 C(T) - C_{\text{final}}^b) \end{cases} \quad (3.15)$$

L'équation (3.15) est appelée équation adjointe (ou modèle adjoint) et la variable P est l'état adjoint de C . Ainsi si l'état C est solution de (3.1) et l'état adjoint P est solution de (3.15), le gradient de la fonction coût vaut

$$\frac{dJ}{dK} = \frac{\partial \mathcal{L}}{\partial K}(C(K), K; P). \quad (3.16)$$

On pose $\nabla_K J = \frac{\partial J}{\partial K}$ et on déduit de (3.9) et (3.11) que

$$\nabla_K J = - \int_0^T \left(\frac{\partial F}{\partial K} \right)^* P dt. \quad (3.17)$$

Le gradient de J est donné par la formule (3.17) et on constate qu'il se calcule par intégration rétrograde du modèle adjoint (équation (3.15)).

Remarque 3.3 *On a identifié les espaces de Hilbert \mathcal{H} , \mathcal{K} et \mathcal{O} avec leurs duaux.*

Remarque 3.4 *Si on avait voulu identifier la condition initiale et/ou les conditions aux limites (comme c'est le cas en météorologie ou en océanographie, voir par exemple [66]), on aurait procédé exactement de la même manière en calculant un ou deux gradient(s) supplémentaire(s).*

La solution K^* du problème (3.3) est donnée en résolvant le système suivant

$$\left\{ \begin{array}{l} \frac{dC}{dt} = F(C(t), K), \quad t \in [0, T], \quad C \in \mathcal{H} \\ C(0) = U \\ -\frac{dP}{dt} = \left(\frac{\partial F}{\partial C} \right)^* P - 2\alpha M_1^*(M_1 C - C^o), \quad t \in [0, T] \\ -P(T) = 2\beta M_2^*(M_2 C(T) - C_{\text{final}}^b) \\ \nabla_K J = 0 \end{array} \right. \quad (3.18)$$

appelé système d'optimalité.

Remarque 3.5 *En pratique on ne résout pas directement le système d'optimalité (3.18), mais on utilise un algorithme de descente dans la direction du gradient.*

Les domaines d'application de ces méthodes de contrôle optimal à des problèmes d'identification pour des systèmes régis par des edp sont multiples et variés. On peut citer en assimilation de données pour la météorologie et l'océanographie Auroux & Blum ([12]), Blum et al. (article dans Handbook of Numerical Analysis [94]), Le Dimet et Talagrand ([68]), Courtier et Talagrand ([43]), Luong ([72]), Blum ([20]) pour la physique des plasmas, De Marsily ([45]) en hydrogéologie, Chavent ([32, 33, 34]) et Tarantola ([93]) pour les méthodes d'inversion, Cartalade et al. ([29, 30]), Mc Laughlin et al. ([77]), Sun ([90]) et Carrera ([28]) pour les écoulements et le transport en milieu poreux, Plessix et al. ([82]) pour la géophysique interne.

Remarque 3.6 *Cette méthode de calcul pour obtenir l'état adjoint est dite **lagrangienne**.*

3.2.3.2 Approche “linéarisée”

Une autre approche est possible, elle utilise le modèle linéaire tangent déduit du problème direct (cf. section 3.2.2). Ces deux approches sont totalement identiques et conduisent aux mêmes résultats. Pour obtenir l'équation adjointe de (3.1), on introduit l'état adjoint P et on prend le produit scalaire de (3.7) avec P , on somme sur l'intervalle $[0, T]$ et après intégration par parties (du même genre que (3.13)) et identification des linéarités par rapport à \tilde{K} dans (3.8), on retrouve que P est solution de (3.15) et que le gradient de J est donné par (3.17) c'est-à-dire que l'état adjoint du problème non linéaire est le même que l'état adjoint du problème linéarisé.

3.2.3.3 Régularisation

Le problème inverse inverse étant instable, une technique courante consiste à le régulariser. En pratique, on rajoute un terme à la fonction coût, par exemple on peut prendre un terme contenant la différence au sens des moindres carrés entre les paramètres calculés et une ébauche de ces paramètres (puisqu'en pratique on ne connaît pas les paramètres observés). Ce terme agit comme un terme de régularisation au sens de Tikhonov [97] (c'est la méthode la plus classique pour régulariser). On peut aussi imaginer un terme de régularisation portant non plus sur les paramètres eux mêmes mais sur leurs dérivées afin d'avoir une certaine régularité sur ces fonctions. On peut également mentionner l'approche multi-échelle ou multi-grille [39, 71], la régularisation au sens BV (pour conserver les discontinuités), au sens de la norme L^2 du gradient ou des normes spécifiquement appropriées au problème physique concerné (norme de la divergence pour un problème de méca fluide, etc.). On peut se référer à [35] pour une étude faisant le lien entre les différentes techniques de régularisation et la notion de problème quadratiquement bien posé.

Ainsi, en suivant la même démarche que précédemment, on constate que si l'on rajoute des termes de régularisation à la fonction coût, le système adjoint associé à cette nouvelle fonctionnelle reste inchangé. Par contre, le gradient lui prend en compte ces nouveaux termes régularisants.

Dans la suite, nous ne prendrons pas en compte le terme de régularisation. En effet, dans notre cas les paramètres sont constants par zones, il n'y a pas d'oscillations possibles et donc pas de terme de régularisation à rajouter pour lisser la solution [48].

3.2.4 Obtention formelle de l'équation de transport adjointe

Nous voulons identifier les paramètres de transport (diffusion et porosité) du modèle mathématique (1.6) connaissant une distribution de la solution en différents points de l'espace et à différents temps de simulation. Nous allons ci-dessous formuler le problème de contrôle optimal et écrire les équations adjointes pour les deux cas-tests définis au chapitre 2. Afin d'obtenir l'état adjoint, nous utiliserons la méthode dite lagrangienne exposée dans la section 3.2.3.1.

3.2.4.1 Cas-test “colis”

On rappelle les notations introduites dans la section 2.6.1 et on se réfère à la figure 3.1 :

- * $\Omega = \Omega_1 \cup \Omega_2$ et $\partial\Omega = \Gamma_N \cup \Gamma_S \cup \Gamma_D$,
- * γ désigne l'interface entre les deux domaines Ω_1 et Ω_2 ,
- * Σ désigne le domaine dans lequel on mesure la concentration à l'instant final,
- * le terme source S dans l'équation (1.6) est nul,
- * on prend comme conditions initiales et aux limites (2.28)-(2.29), *i.e.*

$$\begin{cases} C(t=0, x) = 0 & \text{sur } \Omega_1 \\ C(t=0, x) = 0 & \text{sur } \Omega_2 \end{cases} \quad \begin{cases} C(t, x) = 0 & \text{sur } \Gamma_N \\ D\nabla C(t, x).n = 0 & \text{sur } \Gamma_S \\ D\nabla C(t, x).n = q_D(t, x) & \text{sur } \Gamma_D \end{cases}$$

avec les conditions de transmission suivantes

$$(D\nabla C.n_1)|_\gamma + (D\nabla C.n_2)|_\gamma = 0 \quad (3.19)$$

$$(q.n_1)|_\gamma + (q.n_2)|_\gamma = 0 \quad (3.20)$$

où n_1 est le vecteur normal unitaire extérieur à Ω_1 et n_2 est le vecteur normal unitaire extérieur à Ω_2 . Sur γ , nous avons $n_1 = -n_2$ et donc la quantité $D\nabla C.n$ est continue à l'interface. De même, la concentration C est continue à l'interface.

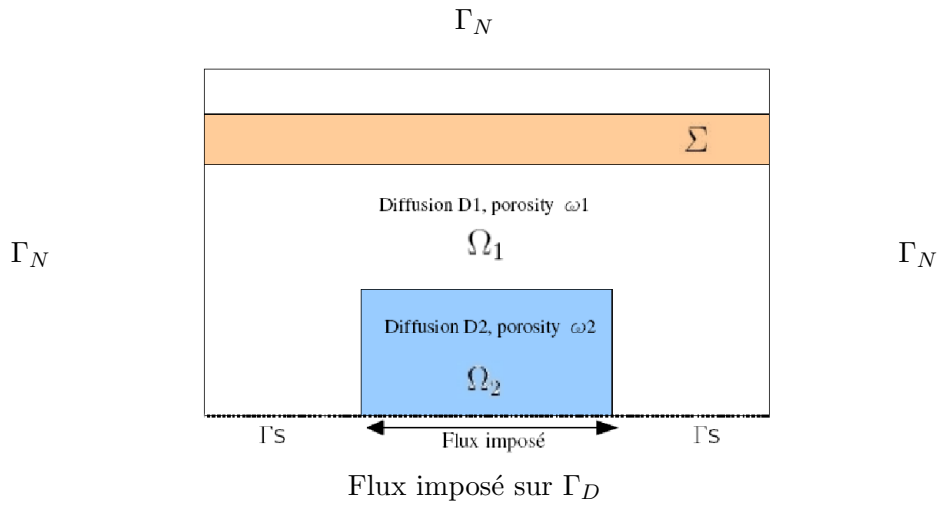


FIG. 3.1 – Domaine Ω

La fonction coût J est définie par

$$J(C(\omega, D)) = \underbrace{\alpha \int_0^T \int_\gamma [C(t, x) - C_{obs}(t, x)]^2 d\sigma dt}_{(\pi)} + \underbrace{\beta \int_\Sigma [C(T, x) - C_{obs}(T, x)]^2 dx}_{(\kappa)} \quad (3.21)$$

où α et β sont des poids qu'on ajuste pour que les deux termes (π) et (κ) de la fonction coût soient du même ordre. Le terme (π) correspond à la partie intégrée en temps et le terme (κ) à l'instant final.

Le Lagrangien de notre problème est donné par

$$\begin{aligned} \mathcal{L}((\omega, D), C, P) = & J(C) + \int_0^T \int_\Omega \omega R \partial_t C P dx dt - \underbrace{\int_0^T \int_\Omega \nabla \cdot (D \nabla C) P dx dt}_{(\lambda)} \\ & + \underbrace{\int_0^T \int_\Omega \nabla \cdot (q C) P dx dt}_{(\mu)} + \int_0^T \int_\Omega \lambda \omega R P C dx dt. \end{aligned} \quad (3.22)$$

On découpe les termes (λ) et (μ) sur Ω_1 et Ω_2

$$-\int_0^T \int_{\Omega} \nabla \cdot (D \nabla C) P \, dx \, dt = -\int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla C) P \, dx \, dt - \int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla C) P \, dx \, dt \quad (3.23)$$

$$\int_0^T \int_{\Omega} \nabla \cdot (q C) P \, dx \, dt = \int_0^T \int_{\Omega_1} \nabla \cdot (q C) P \, dx \, dt + \int_0^T \int_{\Omega_2} \nabla \cdot (q C) P \, dx \, dt. \quad (3.24)$$

On calcule la partie droite de (3.23)

$$\begin{aligned} -\int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla C) P \, dx \, dt &= \int_0^T \int_{\Omega_1} D (\nabla C \cdot \nabla P) \, dx \, dt - \int_0^T \int_{\Gamma_N \cup \Gamma_S \cup \gamma} D \nabla C \cdot n_1 P \, d\sigma \, dt \\ &= -\int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla P) C \, dx \, dt - \int_0^T \int_{\Gamma_N \cup \Gamma_S \cup \gamma} D \nabla C \cdot n_1 P \, d\sigma \, dt \\ &\quad + \int_0^T \int_{\Gamma_N \cup \Gamma_S \cup \gamma} D \nabla P \cdot n_1 C \, d\sigma \, dt \\ &= -\int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\Gamma_S} D \nabla P \cdot n_1 C \, d\sigma \, dt \\ &\quad - \int_0^T \int_{\gamma} D (\nabla C \cdot n_1 P - \nabla P \cdot n_1 C) \, d\sigma \, dt \\ &\quad - \int_0^T \int_{\Gamma_N} D \nabla C \cdot n_1 P \, d\sigma \, dt. \end{aligned} \quad (3.25)$$

De même

$$\begin{aligned} -\int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla C) P \, dx \, dt &= -\int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla P) C \, dx \, dt \\ &\quad - \int_0^T \int_{\Gamma_D} (P q_D - D \nabla P \cdot n_2 C) \, d\sigma \, dt \\ &\quad - \int_0^T \int_{\gamma} D (\nabla C \cdot n_2 P - \nabla P \cdot n_2 C) \, d\sigma \, dt. \end{aligned} \quad (3.26)$$

Donc le terme (3.23) devient

$$\begin{aligned} -\int_0^T \int_{\Omega} \nabla \cdot (D \nabla C) P \, dx \, dt &= -\int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\Gamma_S} D \nabla P \cdot n_1 C \, d\sigma \, dt \\ &\quad - \int_0^T \int_{\Gamma_D} (P q_D - D \nabla P \cdot n_2 C) \, d\sigma \, dt - \int_0^T \int_{\Gamma_N} D \nabla C \cdot n_1 P \, d\sigma \, dt \\ &\quad + \int_0^T \int_{\gamma} (D \nabla P \cdot n_1 + D \nabla P \cdot n_2) C \, d\sigma \, dt. \end{aligned} \quad (3.27)$$

On calcule la partie droite de (3.24)

$$\begin{aligned} \int_0^T \int_{\Omega_1} \nabla \cdot (q C) P \, dx \, dt &= -\int_0^T \int_{\Omega_1} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_N \cup \Gamma_S \cup \gamma} q \cdot n_1 P C \, d\sigma \, dt \\ &= -\int_0^T \int_{\Omega_1} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_S \cup \gamma} q \cdot n_1 P C \, d\sigma \, dt. \end{aligned} \quad (3.28)$$

De même

$$\int_0^T \int_{\Omega_2} \nabla \cdot (qC) P \, dx \, dt = - \int_0^T \int_{\Omega_2} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_D \cup \gamma} q \cdot n_2 P C \, d\sigma \, dt. \quad (3.29)$$

Donc le terme (3.24) devient

$$\begin{aligned} \int_0^T \int_{\Omega} \nabla \cdot (qC) P \, dx \, dt &= - \int_0^T \int_{\Omega} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\gamma} (q \cdot n_1 + q \cdot n_2) P C \, d\sigma \, dt \\ &+ \int_0^T \int_{\Gamma_D} q \cdot n_2 P C \, d\sigma \, dt + \int_0^T \int_{\Gamma_S} q \cdot n_1 P C \, d\sigma \, dt. \end{aligned} \quad (3.30)$$

Le Lagrangien (3.22) s'écrit donc

$$\begin{aligned} \mathcal{L}((\omega, D), C, P) &= J(C) - \int_0^T \int_{\Omega} \omega R \partial_t P C \, dx \, dt + \int_{\Omega} \omega R P(T, x) C(T, x) \, dx \\ &- \int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\gamma} (D \nabla P \cdot n_1 + D \nabla P \cdot n_2) C \, d\sigma \, dt \\ &+ \int_0^T \int_{\Gamma_S} D \nabla P \cdot n_1 C \, d\sigma \, dt - \int_0^T \int_{\Gamma_N} D \nabla C \cdot n_1 P \, d\sigma \, dt \\ &- \int_0^T \int_{\Gamma_D} (P q_D(t, x) - D \nabla P \cdot n_2 C) \, d\sigma \, dt - \int_0^T \int_{\Omega} q \cdot \nabla P C \, dx \, dt \\ &+ \int_0^T \int_{\Gamma_D} q \cdot n_2 P C \, d\sigma \, dt + \int_0^T \int_{\Gamma_S} q \cdot n_1 P C \, d\sigma \, dt \\ &+ \int_0^T \int_{\Omega} \lambda \omega R P C \, dx \, dt. \end{aligned} \quad (3.31)$$

On prend l'état adjoint P nul sur Γ_N , on dérive le Lagrangien (3.31).

$$\begin{aligned} \left(\frac{\partial \mathcal{L}}{\partial C}, \tilde{C} \right) &= \left(\frac{\partial J}{\partial C}, \tilde{C} \right) - \int_0^T \int_{\Omega} \omega R \partial_t P \tilde{C} \, dx \, dt + \int_{\Omega} \omega R P(T, x) \tilde{C}(T, x) \, dx \\ &- \int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) \tilde{C} \, dx \, dt + \int_0^T \int_{\gamma} (D \nabla P \cdot n_1 + D \nabla P \cdot n_2) \tilde{C} \, d\sigma \, dt \\ &+ \int_0^T \int_{\Gamma_S} D \nabla P \cdot n_1 \tilde{C} \, d\sigma \, dt + \int_0^T \int_{\Gamma_D} D \nabla P \cdot n_2 \tilde{C} \, d\sigma \, dt \\ &- \int_0^T \int_{\Omega} q \cdot \nabla P \tilde{C} \, dx \, dt + \int_0^T \int_{\Gamma_D} q \cdot n_2 P \tilde{C} \, d\sigma \, dt \\ &+ \int_0^T \int_{\Gamma_S} q \cdot n_1 P \tilde{C} \, d\sigma \, dt + \int_0^T \int_{\Omega} \lambda \omega R P \tilde{C} \, dx \, dt. \end{aligned} \quad (3.32)$$

La Gâteaux-dérivée de J par rapport à C vaut

$$\left(\frac{\partial J}{\partial C}, \tilde{C} \right) = 2\alpha \int_0^T \int_{\gamma} [C(t, x) - C_{obs}(t, x)] \tilde{C}(t, x) \, d\sigma \, dt + 2\beta \int_{\Sigma} [C(T, x) - C_{obs}(T, x)] \tilde{C}(T, x) \, dx. \quad (3.33)$$

Si $C(K)$ est solution de $E(C; K) = 0$, on a alors que

$$J(K) = \mathcal{L}(C(K), P; K) \quad \text{et} \quad \delta J = \frac{\partial \mathcal{L}}{\partial C}(C(K), P; K) \delta C + \frac{\partial \mathcal{L}}{\partial K}(C(K), P; K) \delta K. \quad (3.34)$$

On cherche l'état adjoint P tel que

$$\frac{\partial \mathcal{L}}{\partial C}(C(K), P; K) \delta C = 0 \quad \forall \delta C, \quad (3.35)$$

de sorte que l'on ait

$$\frac{\partial J}{\partial K}(K) = \frac{\partial \mathcal{L}}{\partial K}(C(K), P; K). \quad (3.36)$$

Il vient alors

$$-\omega R \partial_t P - \nabla \cdot (D \nabla P) - q \nabla P + \lambda \omega R P = 0 \quad (3.37)$$

avec pour conditions finales et aux limites

$$\left\{ \begin{array}{ll} P(t, x) & = 0 \text{ sur } \Gamma_N, \\ D \nabla P \cdot n_2 + q \cdot n_2 P & = 0 \text{ sur } \Gamma_D, \\ D \nabla P \cdot n_1 + q \cdot n_1 P & = 0 \text{ sur } \Gamma_S, \\ -(D \nabla P \cdot n_1 + D \nabla P \cdot n_2) & = 2\alpha [C(t, x) - C_{obs}(t, x)] \text{ sur } \gamma, \\ -R \omega P(T, x) & = 2\beta [C(T, x) - C_{obs}(T, x)] \text{ dans } \Sigma, \\ -R \omega P(T, x) & = 0 \text{ dans } \Omega \setminus \Sigma. \end{array} \right. \quad (3.38)$$

En utilisant (3.22) et (3.36), on obtient pour expression des gradients

$$\left\{ \begin{array}{l} \left(\frac{\partial J}{\partial \omega}, \tilde{\omega} \right) = \int_0^T \int_{\Omega} (R \partial_t C + \lambda R C) P \tilde{\omega} dx dt \\ \left(\frac{\partial J}{\partial D}, \tilde{D} \right) = \int_0^T \int_{\Omega} \nabla C \cdot \nabla P \tilde{D} dx dt \end{array} \right. \quad (3.39)$$

$$\left(\frac{\partial J}{\partial D}, \tilde{D} \right) = \int_0^T \int_{\Omega} \nabla C \cdot \nabla P \tilde{D} dx dt \quad (3.40)$$

Une fois le gradient calculé, il ne reste plus qu'à le transmettre à l'optimiseur pour commencer la minimisation. Nous expliquons en détails dans la section 3.3 l'algorithme numérique mis en œuvre et nous présentons les résultats obtenus.

Remarque 3.7 *Les paramètres étant considérés constants par zones, l'intégrale en espace dans les relations (3.39) et (3.40) ne portent plus que sur la zone considérée.*

3.2.4.2 Cas-test "réel"

Nous procédons de la même manière que dans le paragraphe précédent pour obtenir le gradient de la fonction coût par rapport aux paramètres à identifier. On rappelle les notations introduites dans la section 2.6.2 et on se réfère à la figure 3.2 :

- * $\Omega = \Omega_1 \cup \Omega_2 \cup \Omega_3$ et $\partial\Omega = \Gamma = \Gamma_1 \cup \Gamma_2 \cup \Gamma_3$,
- * on note $\partial\Omega_1 = \Gamma_1 \cup \Gamma_{13}$ la frontière de Ω_1 , Γ_1 désigne la frontière extérieure et Γ_{13} l'interface entre Ω_1 et Ω_3 ,
- * on note $\partial\Omega_2 = \Gamma_2 \cup \Gamma_{23}$ la frontière de Ω_2 , Γ_2 désigne la frontière extérieure et Γ_{23} l'interface entre Ω_2 et Ω_3 ,
- * on note $\partial\Omega_3 = \Gamma_3 \cup \Gamma_{13} \cup \Gamma_{23}$ la frontière de Ω_3 , Γ_3 désigne la frontière extérieure,

- ★ les domaines γ_i , $i = 1, 2, 3$ désigne les trois zones où on mesure la concentration à l'instant final*,
- ★ le terme source S dans l'équation (1.6) est nul,
- ★ on prend comme conditions initiales et aux limites (2.30)-(2.31), *i.e.*

$$\begin{cases} C(t=0, x) = 0 & \text{sur } \Omega_1 \cup \Omega_3 \\ C(t=0, x) = 1 & \text{sur } \Omega_2 \end{cases} \quad D\nabla C(t, x).n = 0 \text{ sur } \Gamma$$

avec les conditions de transmission suivantes

$$\begin{cases} (D\nabla C.n_1)|_{\Gamma_{13}} + (D\nabla C.n_3)|_{\Gamma_{13}} = 0 \\ (q.n_1)|_{\Gamma_{13}} + (q.n_3)|_{\Gamma_{13}} = 0 \end{cases} \quad (3.41)$$

$$\begin{cases} (D\nabla C.n_2)|_{\Gamma_{23}} + (D\nabla C.n_3)|_{\Gamma_{23}} = 0 \\ (q.n_2)|_{\Gamma_{23}} + (q.n_3)|_{\Gamma_{23}} = 0 \end{cases} \quad (3.42)$$

où n_1 est le vecteur normal unitaire extérieur à Ω_1 , n_2 est le vecteur normal unitaire extérieur à Ω_2 et n_3 est le vecteur normal unitaire extérieur à Ω_3 . Sur Γ_{13} , nous avons $n_1 = -n_3$ et sur Γ_{23} , $n_2 = -n_3$. Ainsi, la quantité $D\nabla C.n$ (avec n vecteur normal unitaire extérieur à Γ) est continue aux interfaces et il en est de même pour la concentration C .

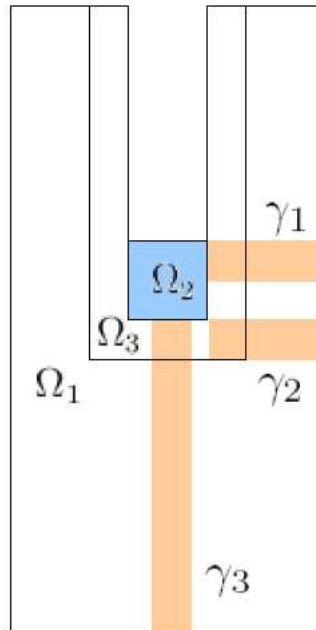


FIG. 3.2 – Domaine Ω

*Pour simplifier l'étude, on a pris dans ce paragraphe pour γ_i ($i = 1, 2, 3$) des zones au lieu de droites comme sur la figure 2.6.

La fonction coût J est définie par

$$J(C(\omega, D)) = \underbrace{\alpha \int_0^T \int_{\Omega_2} [C(t, x) - C_{obs}(t, x)]^2 dx dt}_{(\sigma)} + \underbrace{\beta \sum_{i=1}^3 \int_{\gamma_i} [C(T, x) - C_{obs}(T, x)]^2 dx}_{(\psi)}. \quad (3.43)$$

Comme dans la section précédente, α et β sont des poids qu'on ajuste pour que les deux termes (σ) et (ψ) de la fonction coût soient du même ordre.

Le Lagrangien de notre problème est donné par (3.22) : on découpe les termes (λ) et (μ) sur Ω_1 , Ω_2 et Ω_3

$$\begin{aligned} - \int_0^T \int_{\Omega} \nabla \cdot (D \nabla C) P dx dt &= - \int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla C) P dx dt - \int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla C) P dx dt \\ &\quad - \int_0^T \int_{\Omega_3} \nabla \cdot (D \nabla C) P dx dt \end{aligned} \quad (3.44)$$

$$\begin{aligned} \int_0^T \int_{\Omega} \nabla \cdot (q C) P dx dt &= \int_0^T \int_{\Omega_1} \nabla \cdot (q C) P dx dt + \int_0^T \int_{\Omega_2} \nabla \cdot (q C) P dx dt \\ &\quad + \int_0^T \int_{\Omega_3} \nabla \cdot (q C) P dx dt. \end{aligned} \quad (3.45)$$

On calcule la partie droite de (3.44)

$$\begin{aligned} - \int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla C) P dx dt &= \int_0^T \int_{\Omega_1} D (\nabla C \cdot \nabla P) dx dt - \int_0^T \int_{\Gamma_1 \cup \Gamma_{13}} D \nabla C \cdot n_1 P d\sigma dt \\ &= - \int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla P) C dx dt - \int_0^T \int_{\Gamma_1 \cup \Gamma_{13}} D \nabla C \cdot n_1 P d\sigma dt \\ &\quad + \int_0^T \int_{\Gamma_1 \cup \Gamma_{13}} D \nabla P \cdot n_1 C d\sigma dt \\ &= - \int_0^T \int_{\Omega_1} \nabla \cdot (D \nabla P) C dx dt + \int_0^T \int_{\Gamma_1} D \nabla P \cdot n_1 C d\sigma dt \\ &\quad - \int_0^T \int_{\Gamma_{13}} D (\nabla C \cdot n_1 P - \nabla P \cdot n_1 C) d\sigma dt \end{aligned} \quad (3.46)$$

De même

$$\begin{aligned} - \int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla C) P dx dt &= - \int_0^T \int_{\Omega_2} \nabla \cdot (D \nabla P) C dx dt + \int_0^T \int_{\Gamma_2} D \nabla P \cdot n_2 C d\sigma dt \\ &\quad - \int_0^T \int_{\Gamma_{23}} D (\nabla C \cdot n_2 P - \nabla P \cdot n_2 C) d\sigma dt \end{aligned} \quad (3.47)$$

$$\begin{aligned}
-\int_0^T \int_{\Omega_3} \nabla \cdot (D \nabla C) P \, dx \, dt &= -\int_0^T \int_{\Omega_3} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\Gamma_3} D \nabla P \cdot n_3 C \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{13} \cup \Gamma_{23}} D (\nabla C \cdot n_3 P - \nabla P \cdot n_3 C) \, d\sigma \, dt.
\end{aligned} \tag{3.48}$$

Ainsi le terme (3.44) s'écrit

$$\begin{aligned}
-\int_0^T \int_{\Omega} \nabla \cdot (D \nabla C) P \, dx \, dt &= -\int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\Gamma} D \nabla P \cdot n C \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{13}} D (\nabla P \cdot n_1 C + \nabla P \cdot n_3 C) \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{23}} D (\nabla P \cdot n_2 C + \nabla P \cdot n_3 C) \, d\sigma \, dt.
\end{aligned} \tag{3.49}$$

On calcule la partie droite de (3.45)

$$\int_0^T \int_{\Omega_1} \nabla \cdot (q C) P \, dx \, dt = -\int_0^T \int_{\Omega_1} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_1 \cup \Gamma_{13}} q \cdot n_1 P C \, d\sigma \, dt. \tag{3.50}$$

De même

$$\int_0^T \int_{\Omega_2} \nabla \cdot (q C) P \, dx \, dt = -\int_0^T \int_{\Omega_2} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_2 \cup \Gamma_{23}} q \cdot n_2 P C \, d\sigma \, dt. \tag{3.51}$$

$$\int_0^T \int_{\Omega_3} \nabla \cdot (q C) P \, dx \, dt = -\int_0^T \int_{\Omega_3} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma_3 \cup \Gamma_{13} \cup \Gamma_{23}} q \cdot n_3 P C \, d\sigma \, dt. \tag{3.52}$$

Donc le terme (3.45) devient

$$\int_0^T \int_{\Omega} \nabla \cdot (q C) P \, dx \, dt = -\int_0^T \int_{\Omega} q \cdot \nabla P C \, dx \, dt + \int_0^T \int_{\Gamma} q \cdot n P C \, d\sigma \, dt. \tag{3.53}$$

Le Lagrangien (3.22) s'écrit donc

$$\begin{aligned}
\mathcal{L}((\omega, D), C, P) &= J(C) - \int_0^T \int_{\Omega} \omega R \partial_t P C \, dx \, dt + \int_{\Omega} \omega R P(T, x) C(T, x) \, dx \\
&\quad - \int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) C \, dx \, dt + \int_0^T \int_{\Gamma} D \nabla P \cdot n C \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{13}} D (\nabla P \cdot n_1 C + \nabla P \cdot n_3 C) \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{23}} D (\nabla P \cdot n_2 C + \nabla P \cdot n_3 C) \, d\sigma \, dt - \int_0^T \int_{\Omega} q \cdot \nabla P C \, dx \, dt \\
&\quad + \int_0^T \int_{\Gamma} q \cdot n P C \, d\sigma \, dt + \int_0^T \int_{\Omega} \lambda \omega R P C \, dx \, dt.
\end{aligned} \tag{3.54}$$

D'où

$$\begin{aligned}
\left(\frac{\partial \mathcal{L}}{\partial C}, \tilde{C}\right) &= \left(\frac{\partial J}{\partial C}, \tilde{C}\right) - \int_0^T \int_{\Omega} \omega R \partial_t P \tilde{C} \, dx \, dt + \int_{\Omega} \omega R P(T, x) \tilde{C}(T, x) \, dx \\
&\quad - \int_0^T \int_{\Omega} \nabla \cdot (D \nabla P) \tilde{C} \, dx \, dt + \int_0^T \int_{\Gamma} D \nabla P \cdot n \tilde{C} \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{13}} D (\nabla P \cdot n_1 + \nabla P \cdot n_3) \tilde{C} \, d\sigma \, dt \\
&\quad - \int_0^T \int_{\Gamma_{23}} D (\nabla P \cdot n_2 + \nabla P \cdot n_3) \tilde{C} \, d\sigma \, dt - \int_0^T \int_{\Omega} q \cdot \nabla P \tilde{C} \, dx \, dt \\
&\quad + \int_0^T \int_{\Gamma} q \cdot n P \tilde{C} \, d\sigma \, dt + \int_0^T \int_{\Omega} \lambda \omega R P \tilde{C} \, dx \, dt. \tag{3.55}
\end{aligned}$$

La Gâteaux-dérivée de J par rapport à C vaut

$$\left(\frac{\partial J}{\partial C}, \tilde{C}\right) = 2\alpha \int_0^T \int_{\Omega_2} [C(t, x) - C_{obs}(t, x)] \tilde{C}(t, x) \, dx \, dt + 2\beta \sum_{i=1}^3 \int_{\gamma_i} [C(T, x) - C_{obs}(T, x)] \tilde{C}(T, x) \, dx. \tag{3.56}$$

L'équation adjointe s'exprime sous la forme

$$-\omega R \partial_t P - \nabla \cdot (D \nabla P) - q \cdot \nabla P + \lambda \omega R P = \begin{cases} 0 & \text{dans } \Omega_1 \cup \Omega_3, \\ -2\alpha(C - C_{obs}) & \text{dans } \Omega_2. \end{cases} \tag{3.57}$$

avec pour conditions finales et aux limites

$$\left\{ \begin{array}{l} D \nabla P \cdot n + q \cdot n P = 0 \text{ sur } \Gamma, \\ D \nabla P \cdot n_1 + D \nabla P \cdot n_3 = 0 \text{ sur } \Gamma_{13}, \\ D \nabla P \cdot n_2 + D \nabla P \cdot n_3 = 0 \text{ sur } \Gamma_{23}, \\ -R \omega P(T, x) = 2\beta [C(T, x) - C_{obs}(T, x)] \text{ dans } \gamma_1 \cup \gamma_2 \cup \gamma_3, \\ -R \omega P(T, x) = 0 \text{ dans } \Omega \setminus (\gamma_1 \cup \gamma_2 \cup \gamma_3). \end{array} \right. \tag{3.58}$$

On obtient pour expression des gradients

$$\left\{ \begin{array}{l} \left(\frac{\partial J}{\partial \omega}, \tilde{\omega}\right) = \int_0^T \int_{\Omega} (R \partial_t C + \lambda R C) P \tilde{\omega} \, dx \, dt \end{array} \right. \tag{3.59}$$

$$\left\{ \begin{array}{l} \left(\frac{\partial J}{\partial D}, \tilde{D}\right) = \int_0^T \int_{\Omega} \nabla C \cdot \nabla P \tilde{D} \, dx \, dt \end{array} \right. \tag{3.60}$$

Remarque 3.8 *On remarque que les calculs de l'adjoint dans les deux cas-tests sont similaires et qu'on obtient la même expression pour les gradients. Par contre, les conditions finales et aux limites n'étant pas les mêmes dans les deux cas-tests, on constate que les équations adjointes (3.57)-(3.58) et (3.57)-(3.58) sont différentes (on note en particulier l'apparition d'un terme source dans l'équation adjointe (3.57)).*

Comme nous l'avons déjà signalé dans la section 3.2.2, la méthode de l'état adjoint prouve son efficacité lorsque le nombre de paramètres devient important. Dans les cas-tests présentés, nous avons au plus 6 paramètres à identifier. Cependant, la méthode mise en place est indépendante du nombre de paramètres et permet donc de traiter des problèmes avec un grand nombre de paramètres en résolvant seulement deux systèmes d'edp. Elle trouvera pleinement sa justification dans la partie sur l'identification de zones (cf. chapitre 6).

3.3 Etude numérique du problème inverse

3.3.1 Introduction

Pour résoudre numériquement le problème de contrôle optimal (3.18) afin de reconstituer les paramètres de transport, nous avons pour la minimisation le choix entre utiliser des algorithmes sans gradients et des algorithmes avec gradients. Les algorithmes sans gradients (algorithmes génétiques, simplexe, etc.) permettent de minimiser des fonctionnelles dont la forme peut être très complexe. Notamment, ces méthodes ne s'arrêtent pas dans des minima locaux mais permettent d'obtenir un minimum global de la fonction coût. En revanche, ce type d'approche nécessite un grand nombre d'évaluations de la fonction coût qui peuvent vite devenir coûteux en temps de calcul si cette dernière est issue d'un calcul numérique. C'est pour cette raison que dans la suite nous utiliserons la technique de minimisation au sens des moindres carrés qui requiert le calcul du gradient de la fonction coût.

La plupart de ces algorithmes de minimisation (sans contraintes) reposent sur une méthode dite de descente : en partant d'un point quelconque de l'espace, l'algorithme construit une suite de points dont le but est de converger vers le contrôle optimal en faisant diminuer le plus rapidement possible la fonction coût calculée en ces points. La direction liant deux points successifs est appelée direction de descente. Cette direction est calculée par l'algorithme grâce aux directions du gradient et aux valeurs de la fonction coût évaluée en différents points. Ces algorithmes requièrent à chaque itération le calcul du gradient de la fonction coût par rapport aux paramètres de contrôle.

3.3.2 Le code adjoint

A ce stade du manuscrit, le lecteur pourrait s'attendre, après avoir écrit les équations adjointes continues (3.37) et (3.57) du modèle direct (1.6) pour les deux cas-tests définis dans la section précédente, à ce que l'on discrétise ces équations adjointes en vue de leur implémentation informatique. Ce n'est pas l'approche que nous avons choisi. En effet, en pratique les équations adjointes continues ont peu d'utilité et de plus "coder à la main" ces équations est une tâche longue et fastidieuse. De plus, si l'on ne prend pas l'adjoint du système direct discrétisé, on risque d'observer une divergence de l'algorithme de minimisation. C'est pourquoi, nous avons choisi de différentier automatiquement le code Traces afin d'obtenir directement le code adjoint.

Le code adjoint a donc été entièrement obtenu par différentiation automatique du code direct Traces avec le logiciel Tapenade [92] ce qui a notamment permis d'automatiser la procédure d'obtention du gradient. Nous avons également opté pour cette stratégie car cela nous permet de calculer le gradient de façon très **précise** ce qui est nécessaire pour la résolution de notre problème inverse. Une fois le code différentié, nous aurons accès au gradient de la fonction coût que nous transmettrons ensuite à l'algorithme de minimisation. Nous reviendrons plus longuement sur cette partie dans le chapitre suivant.

3.3.3 Rappels sur les méthodes d'optimisation

La fonctionnelle J et ses dérivées calculées par Tapenade sont ensuite injectées dans un optimiseur qui permet d'obtenir au cours d'une itération donnée des valeurs des paramètres qui rendent la fonctionnelle J inférieure à celle calculée à l'itération précédente. Ainsi en notant x_k le vecteur des paramètres à l'itération k , l'optimiseur fournit à l'itération $k + 1$ la valeur x_{k+1} telle que

$$J(C(x_{k+1})) \leq J(C(x_k)).$$

Pour générer un nouveau point x_{k+1} à partir du point courant x_k , nous cherchons tout d'abord une direction de descente d_k (avec le gradient) à partir de l'itération courante puis on effectue une recherche linéaire qui consiste à chercher le long de cette direction une nouvelle itération pour laquelle la valeur de la fonctionnelle est plus petite. L'algorithme génère un nombre limité de longueur de pas α_k le long de cette direction jusqu'à ce qu'il en trouve un qui approxime le minimum de la relation suivante

$$J(x_k + \alpha_k d_k) = \inf_{\mu \in \mathbb{R}} J(x_k + \mu d_k). \quad (3.61)$$

Au nouveau point x_{k+1} , une nouvelle direction de recherche et une nouvelle longueur de pas sont calculées et le processus est répété. Les algorithmes d'optimisation se distinguent par le choix de la direction et du pas, *i.e.* en choisissant d_k et α_k (le pas) à l'itération k dans la relation

$$x_{k+1} = x_k + \alpha_k d_k. \quad (3.62)$$

Dans la plupart des algorithmes de minimisation, la direction de descente d_k s'écrit sous la forme

$$d_k = -B_k^{-1} \nabla J_k \quad (3.63)$$

où B_k est une matrice symétrique et non singulière. Dans le cas de la méthode de Newton, B_k est exactement la matrice Hessienne $\nabla^2 J_k$. Les méthodes qui utilisent la direction de Newton ont un taux rapide de convergence locale. Le principal inconvénient de cette méthode réside dans le calcul explicite de la matrice des dérivées secondes qui est souvent une étape coûteuse et peu précise tandis que dans les méthodes de type quasi-Newton, B_k est une approximation de la matrice Hessienne qui est mise à jour à chaque itération. Ces méthodes ont l'avantage de ne pas calculer la Hessienne et de converger rapidement. Les mises à jour utilisent le fait que les gradients fournissent une information sur la dérivée seconde de J le long de cette direction ainsi que la formule suivante

$$\nabla^2 J_{k+1}(x_{k+1} - x_k) \approx \nabla J_{k+1} - \nabla J_k. \quad (3.64)$$

Les algorithmes de quasi-Newton évitent la factorisation de B_k à chaque itération en mettant à jour son inverse W_k . Le calcul de d_k est alors effectué en utilisant $d_k = -W_k \nabla J_k$, ce qui peut être implémenté en effectuant une multiplication matrice-vecteur qui est plus simple à réaliser que la mise en œuvre de la formule (3.63).

Cependant, les méthodes de quasi-Newton ne sont pas directement applicables aux problèmes dans lesquels interviennent de nombreux paramètres à cause de l'approximation de la Hessienne (ou de son inverse), habituellement dense, qui augmente la taille mémoire et le temps de calcul. Quand le nombre de paramètres est élevé, on utilise des méthodes de quasi-Newton à mémoire limitée qui ont l'avantage d'être peu coûteuses, robustes et faciles à implémenter. Les approximations des Hessiennes peuvent être alors mémorisées sous formes compactes de quelques vecteurs de longueur n (le nombre d'inconnues du problème).

3.3.4 Description de l'algorithme de minimisation

Dans l'optique d'utiliser des méthodes de quasi-Newton à mémoire limitée, nous avons essayé l'algorithme de minimisation M1QN3 de Gilbert & Lemaréchal [54] de l'INRIA. Cet algorithme utilise la formule BFGS (3.66) (resp. BFGS inverse (3.67)) due à Broyden, Fletcher, Goldfarb et Shanno qui fournit une façon d'évaluer récursivement une approximation du Hessien (resp. de l'inverse du Hessien) au point x_k . Cependant, les essais n'ont pas été concluants car le minimiseur autorisait des points tests x_k où les coefficients pouvaient être négatifs. Pour pouvoir minimiser sous contrainte de positivité des coefficients, nous nous sommes tournés vers l'algorithme N2QN1 de Lemaréchal & Panier [69] qui est un optimiseur avec contraintes de bornes et avec lequel nous avons eu de meilleurs résultats. N2QN1 utilise une méthode de quasi-Newton, dite de BFGS et une stratégie de contraintes actives dite à la Rosen-Goldfarb. La méthode de type quasi-Newton est en réalité une méthode de type gradient car elle utilise seulement les variations du gradient de la fonction coût J à minimiser entre deux itérés successifs pour obtenir l'information sur le Hessien.

On rappelle qu'on veut résoudre le problème de minimisation (3.3), *i.e.*

$$\begin{cases} \min J(x) \\ b_{\text{inf}} \leq x \leq b_{\text{sup}} \end{cases} \quad (3.65)$$

où $b_{\text{inf}} \in \mathbb{R}^N$ et $b_{\text{sup}} \in \mathbb{R}^N$ sont les bornes. On note I l'ensemble des indices des bornes jugées actives au point courant ($i \in I$ si $x_i \simeq (b_{\text{inf}})_i$ ou $x_i \simeq (b_{\text{sup}})_i$). On désignera par $x_k = (\omega_k, D_k)$ le point courant, B_k est l'approximation courante de l'Hessien $\nabla^2 J(x_k)$, H_k est l'approximation courante de l'inverse de l'Hessien $\nabla^2 J(x_k)^{-1}$ et $G_k = \nabla J(x_k)$ est le gradient de la fonction coût J au point x_k , calculée pour le produit scalaire $\langle \cdot, \cdot \rangle$. Enfin, on pose $s_k = x_{k+1} - x_k$ et $y_k = G_{k+1} - G_k$. On rappelle les formules de BFGS et BFGS inverse ($\overline{\text{BFGS}}$)

$$\text{BFGS}(B_{k+1}, y_k, s_k) = B_k - \frac{(B_k s_k) \otimes (B_k s_k)}{\langle s_k, B_k s_k \rangle} + \frac{y_k \otimes y_k}{\langle y_k, s_k \rangle} \quad (3.66)$$

$$\overline{\text{BFGS}}(H_{k+1}, y_k, s_k) = \left(I - \frac{s_k \otimes y_k}{\langle y_k, s_k \rangle} \right) H_k \left(I - \frac{y_k \otimes s_k}{\langle y_k, s_k \rangle} \right) + \frac{s_k \otimes s_k}{\langle y_k, s_k \rangle} \quad (3.67)$$

avec $s, y \in \mathbb{R}^m$ et \otimes l'opérateur linéaire défini par

$$\begin{aligned} \otimes : \mathbb{R}^m \times \mathbb{R}^m &\longrightarrow \mathcal{L}(\mathbb{R}^m) \\ (s, y) &\longmapsto [s \otimes y] \end{aligned}$$

qui à tout $w \in \mathbb{R}^m$ associe $[s \otimes y].w = \langle y, w \rangle s$.

On donne aussi la formule du gradient projeté, dont la $i^{\text{ème}}$ composante vaut

$$\begin{cases} (\nabla J(x))_i & \text{si } (b_{\text{inf}})_i < x_i < (b_{\text{sup}})_i \\ (\nabla J(x))_i^- & \text{si } x_i = (b_{\text{inf}})_i \\ (\nabla J(x))_i^+ & \text{si } x_i = (b_{\text{sup}})_i \end{cases} \quad (3.68)$$

où pour $f \in \mathbb{R}$, $f^+ = \max(0, f)$ et $f^- = \max(0, -f)$.

On peut résumer l'algorithme de minimisation par le schéma suivant

Algorithme 3.1 Minimisation

- 1: **Données** : point de départ x_0
- 2: **A chaque itération** de l'algorithme, le point $x_k \in \mathbb{R}^m$ est connu et on effectue
- 3: calcul de la direction de déplacement d_k en x_k en résolvant le problème

$$\begin{cases} \min \nabla J(x_k)^T d_k + \frac{1}{2} d_k^T B_k d_k \\ d_i = 0, \text{ pour } i \in I \end{cases} \quad (3.69)$$

- 4: recherche linéaire sur le segment réalisable porté par la direction d_k ainsi calculée
- 5: modification de B_k par la formule (3.66).
- 6: Avant de retourner à l'étape 1, on révisé I de la façon suivante
 - si la recherche linéaire a buté sur une nouvelle borne, on insère l'indice de cette borne dans I ;
 - si aucune borne nouvelle n'a été rencontrée, soit on garde le même I , soit on lui retranche un indice de bornes si cela en vaut la peine.

TAB. 3.1 – Algorithme de minimisation

Le but de l'algorithme est de trouver un x_k en lequel le gradient projeté est nul, c'est-à-dire que les conditions de Kuhn-Tucker sont vérifiées.

Critère d'arrêt

N2QN1 s'arrête lorsque la valeur moyenne des composantes projetées du gradient par rapport aux variables adimensionnées x_i/dx_{min_i} (où dx_{min} est un vecteur de taille N dont la valeur est fixée à 10^{-30}) devient inférieure à la variable $epsabs$ que nous avons fixé à 10^{-30} .

Remarque 3.9 On utilise le produit scalaire euclidien $\langle \cdot, \cdot \rangle$ donné par $\langle x, y \rangle = x^T y = \sum_{i=1}^n x_i y_i$ et donc le gradient de J est formé des dérivées partielles $\partial J / \partial x_i$ pour $i = 1, \dots, n$.

Remarque 3.10 Avec le produit scalaire euclidien, la matrice dans la base canonique $(e_i)_{1 \leq i \leq n}$ du produit tensoriel $[u \otimes v]$ s'écrit $[u \cdot v^T]$.

Remarque 3.11 Les formules de type BFGS et BFGS inverse ont la propriété de conserver le caractère défini positif de B_k et H_k si et seulement si $\langle y_k, s_k \rangle$ est positif, ce qui assure que d_k est effectivement une direction de descente.

3.3.5 Implémentation

Comme nous l'avons expliqué dans l'introduction, nous avons dû réadapter le code Traces afin de pouvoir le différentier correctement avec le logiciel Tapenade [92]. Les changements opérés dans le code sont nombreux : réécriture et transformation de certaines parties, implémentation de la fonction coût du problème inverse, création de fichiers de données en binaire pour ne pas avoir d'erreur d'arrondis, programmation des routines et des interfaces de résolution du problème d'estimation de paramètres et d'identification de

zones, etc. Une partie de ces modifications a pu être faite grâce à Tapenade qui s'est révélé être un très bon débogueur. De plus, les échanges avec l'équipe Tropics de l'INRIA Sophia-Antipolis ont permis de faire nettement évoluer le logiciel de différentiation. Une fois cette phase de développement terminée, nous avons intégré au code le minimiseur N2QN1 de telle sorte qu'une fois la fonction coût calculée par Traces et l'adjoint obtenu avec Tapenade, nous en déduisons la valeur du gradient qui est, avec la valeur de la fonction coût, directement injectée dans N2QN1. Un des avantages du code ainsi remanié est sa capacité à traiter aussi bien le problème d'estimation de paramètres que le problème d'identification de zones.

Chapitre 4

Différentiation automatique

Sommaire

4.1	Généralités sur le calcul des dérivées	58
4.1.1	Introduction	58
4.1.2	Les différentes méthodes de calcul des dérivées	58
4.1.2.1	Différences finies	59
4.1.2.2	Méthode de la variable complexe	61
4.1.2.3	Linéarisation	61
4.1.2.4	Méthode de l'état adjoint	62
4.2	Différentiation automatique du code par Tapenade	62
4.2.1	Généralités sur la différentiation automatique	62
4.2.1.1	Principe de la dérivation automatique	63
4.2.1.2	Difficultés	64
4.2.2	Présentation de Tapenade	64
4.2.2.1	Fonctionnalités de Tapenade	65
4.2.2.2	Checkpointing	66
4.2.2.3	Utilisation de Tapenade	67
4.2.2.4	Code adjoint généré par Tapenade	67
4.3	Du point de vue numérique	69
4.3.1	Validation des codes générés automatiquement	69
4.4	Conclusion	70
4.4.1	Résumé	70
4.4.2	Discussion sur la différentiation automatique	70

Dans le chapitre précédent, nous avons vu comment calculer efficacement le gradient en continu par la méthode de l'état adjoint. Nous allons voir ici comment calculer le gradient de la fonction coût numériquement. Il est en effet important de différencier un problème discrétisé et non pas de discrétiser un problème différencié parce que les opérations de discrétisation et de différenciation ne commutent pas. Nous avons pris le parti de ne pas écrire les équations adjointes discrètes, nous retenons juste qu'il est préférable de prendre l'adjoint du système direct discrétisé. Dans le cas contraire, on risque d'observer une divergence de l'algorithme de minimisation. C'est pourquoi nous avons choisi de dériver le code direct par différenciation automatique.

4.1 Généralités sur le calcul des dérivées

4.1.1 Introduction

La différenciation automatique permet de générer automatiquement un programme calculant les dérivées d'une fonction à partir d'un programme informatique. Le code d'optimisation de la fonctionnelle coût nécessite l'écriture de deux procédures : l'une résout le problème direct et calcule la fonction coût (c'est le code `Traces`) et l'autre calcule le gradient de la fonction coût.

Le calcul des dérivées pourrait être effectué par différences finies mais cette méthode est coûteuse en temps surtout quand le nombre de paramètres de contrôle est élevé et le choix de la longueur du pas est difficile. De plus, cette approche ne permet pas de calculer exactement le gradient mais de façon approchée et cela conduit à une optimisation peu précise du problème. En effet, pour une bonne convergence des algorithmes d'optimisation, nous devons calculer exactement le gradient de la fonction coût et nous ne pouvons pas atteindre ce niveau de précision de cette façon. Nous pourrions, en vue d'obtenir le code adjoint, différencier le modèle direct continu, puis le discrétiser et l'implémenter dans un programme. Mais une telle mise en œuvre est plus adaptée pour les petits modèles, car cela nécessite beaucoup de temps.

Nous choisissons donc de calculer exactement le gradient de la fonction coût par différenciation automatique du code source. Cette méthode semble être la meilleure façon de dériver le code rapidement et sans erreur.

Après avoir rappelé les différentes façons de calculer des dérivées, nous donnerons le principe général de la différenciation automatique des programmes. Nous expliquerons ensuite comment nous obtenons le code adjoint en utilisant le générateur automatique de codes `Tapenade` [92] puis le gradient de la fonction coût.

4.1.2 Les différentes méthodes de calcul des dérivées

Nous rappelons ici les différentes techniques pour évaluer la dérivée d'une fonction coût J dépendant de la concentration C du radionucléide (variable d'état) par rapport aux paramètres de contrôle $K = (\omega, D)$, dans le cas où $J(K, C(K))$ dépend de K par l'intermédiaire de la concentration C qui est solution de l'équation aux dérivées partielles (2.4). On peut réécrire l'équation (2.4) soit sous la forme $E(K, C(K)) = 0$ avec

$$E(C; K) := \partial_t(\omega RC) - \nabla \cdot (D \cdot \nabla C - qC) + \lambda \omega RC - S \quad (4.1)$$

soit sous la forme (3.1) avec

$$F(C(t); K) = (\omega R)^{-1} (\nabla \cdot (D \cdot \nabla C - qC) - \lambda \omega RC + S). \quad (4.2)$$

Les formules (4.1) et (4.2) seront utilisées dans la suite pour simplifier les expressions.

4.1.2.1 Différences finies

La méthode des différences finies est sans doute la technique la plus connue pour calculer des dérivées discrètes. Cependant, cette méthode n'est pas exacte. En effet, on a besoin dans notre cas de calculer la dérivée de la fonction $J(K)$ par rapport à K , où $K = (\omega, D) \in \mathbb{R}^N$. Or, pour calculer une des N composantes du vecteur gradient, il nous faut calculer la limite

$$\frac{\partial J}{\partial K_i}(K) = \lim_{\varepsilon \rightarrow 0} \frac{J(K + \varepsilon e_i, C(K + \varepsilon e_i)) - J(K, C(K))}{\varepsilon} \quad (4.3)$$

où e_i est un vecteur de la base canonique de \mathbb{R}^N , ce qui revient à calculer au premier ordre soit (formule des différences finies décentrées à droite)

$$\frac{dJ}{dK_i} \approx \frac{J(K + \varepsilon e_i, C(K + \varepsilon e_i)) - J(K, C(K))}{\varepsilon} \quad (4.4)$$

soit (formule des différences finies décentrées à gauche)

$$\frac{dJ}{dK_i} \approx \frac{J(K, C(K)) - J(K - \varepsilon e_i, C(K - \varepsilon e_i))}{\varepsilon}. \quad (4.5)$$

Afin d'avoir une meilleure approximation de la dérivée, on pourrait utiliser la formule des différences finies centrées à la place de (4.4), *i.e.*

$$\frac{dJ}{dK_i} \approx \frac{J(K + \varepsilon e_i, C(K + \varepsilon e_i)) - J(K - \varepsilon e_i, C(K - \varepsilon e_i))}{2\varepsilon}. \quad (4.6)$$

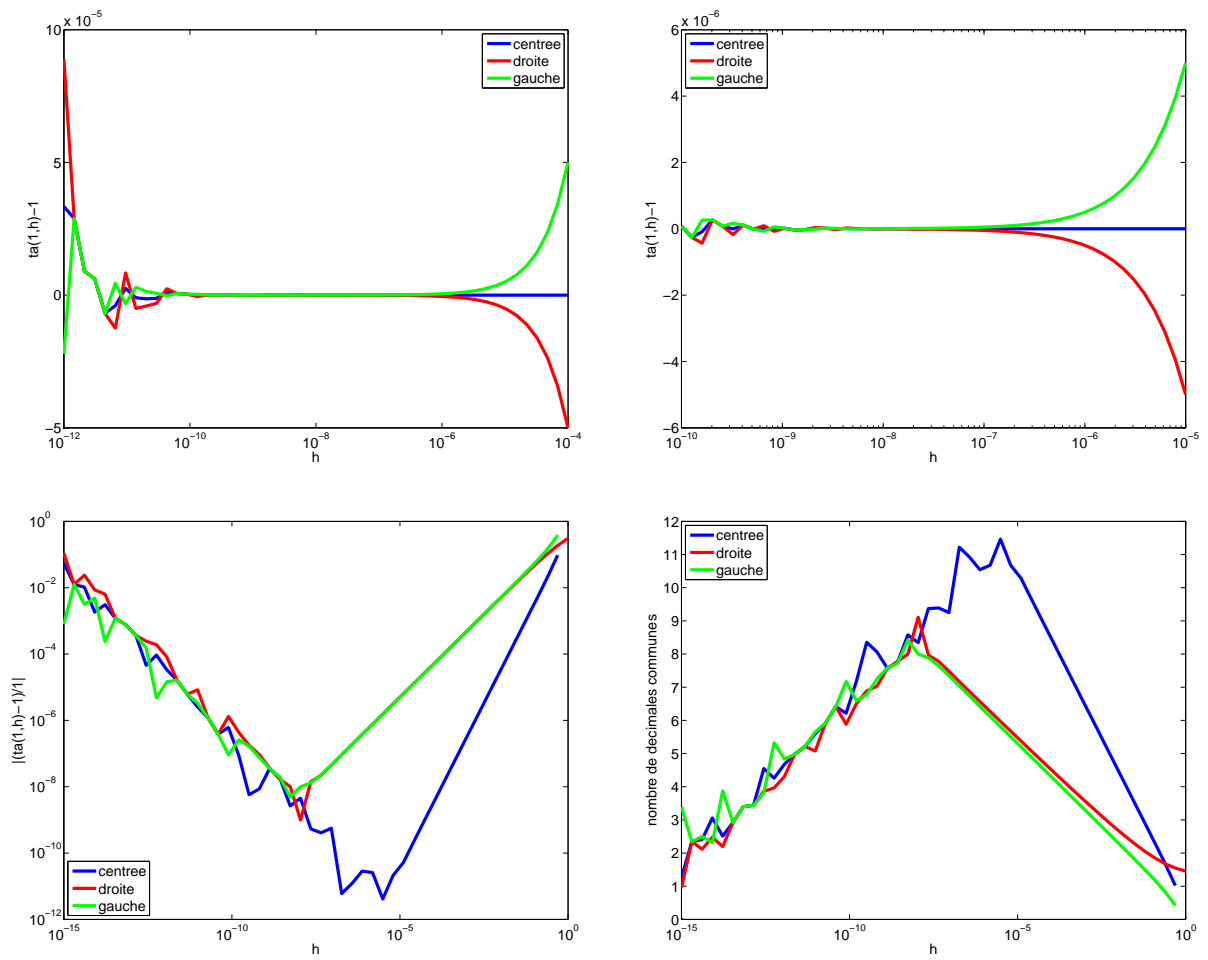
Cependant, ce calcul est très coûteux (le nombre des évaluations est proportionnel à N) et il n'est pas précis. Il faut en effet trouver la valeur optimale de ε pour avoir une valeur précise de la dérivée.

Expériences numériques

Pour bien comprendre les inconvénients liés à la méthode des différences finies, nous calculons les trois rapports (différences finies décentrées à droite et à gauche et différences finies centrées) pour la fonction logarithme au point 1 avec des valeurs de ε de plus en plus petites. On peut voir les résultats sur la figure 4.1.

Les courbes du haut représentent la différence entre le taux d'accroissement (ta) de la fonction logarithme autour du point $(1, \log(1))$ et la valeur exacte de la dérivée de $\log x$ au point 1 en fonction du pas des différences finies compris entre 10^{-12} et 10^{-4} pour la courbe de gauche et entre 10^{-10} et 10^{-5} pour la courbe de droite. La courbe en bas à gauche donne l'évolution de la différence relative entre le taux d'accroissement et la dérivée. La courbe en bas à droite montre le nombre de décimales communes entre le taux d'accroissement et la dérivée de $\log x$ au point 1. Ce nombre est d'abord croissant (on s'approche de la limite) puis décroissant (il devient trop faible devant la précision du calcul). En général, un ordre de 6 à 8 décimales communes est encourageant. Ici, on observe un maximum de 11 (resp. 9) décimales communes pour les différences centrées (resp. décentrées) pour une valeur du pas compris entre 10^{-7} et 10^{-6} (resp. 10^{-9} et 10^{-8}).

On remarque que les courbes présentent à chaque fois un palier, mais il est difficile de prévoir où se trouvera ce palier (sans avoir calculé plusieurs taux d'accroissements) et de dire pour quelle valeur de ε ce palier est optimal. On gardera en tête que les différences divisées ne sont pas le meilleur moyen pour évaluer des dérivées mais restent cependant un bon outil pour vérifier le calcul des dérivées générées automatiquement.

FIG. 4.1 – Différences finies pour $\log x$ au point 1

4.1.2.2 Méthode de la variable complexe

Afin d'éviter les erreurs dues à la soustraction dans (4.4) et (4.6), on peut utiliser la méthode de la variable complexe [89]. La fonction coût J est à valeurs réelles et en supposant qu'elle soit analytique, on remplace dans (4.6) εe_i par $\mathbf{i}\varepsilon$ avec $\mathbf{i} = \sqrt{-1}$ et on écrit

$$J(K + \mathbf{i}\varepsilon, C(K + \mathbf{i}\varepsilon)) = J(K, C(K)) + \mathbf{i}\varepsilon \frac{dJ}{dK} - \frac{\varepsilon^2}{2} \frac{d^2J}{dK^2} - \mathbf{i} \frac{\varepsilon^3}{6} \frac{d^3J}{dK^3} + o(\varepsilon^3). \quad (4.7)$$

En prenant la partie imaginaire des deux côtés de l'égalité (4.7), il vient

$$\frac{\mathcal{I}m[J(K + \mathbf{i}\varepsilon, C(K + \mathbf{i}\varepsilon))]}{\varepsilon} = \frac{dJ}{dK} - \frac{\varepsilon^2}{6} \frac{d^3J}{dK^3} + o(\varepsilon^2) \quad (4.8)$$

c'est-à-dire

$$\frac{\mathcal{I}m[J(K + \mathbf{i}\varepsilon, C(K + \mathbf{i}\varepsilon))]}{\varepsilon} \approx \frac{dJ}{dK}. \quad (4.9)$$

Le coût de cette méthode reste néanmoins proportionnel à N .

Expériences numériques

Si on veut calculer comme à la section 4.1.2.1, la dérivée de la fonction logarithme au point $x_0 = 1$, la formule (4.9) devient

$$\begin{aligned} \frac{d \log(x)}{dx}(x_0) &= \frac{\mathcal{I}m[\log(x_0 + \mathbf{i}\varepsilon)]}{\varepsilon} \\ \frac{d \log(x)}{dx}(x_0) &= \frac{1}{x_0} - \frac{\varepsilon^2}{3x_0^3} + o(\varepsilon^2). \end{aligned} \quad (4.10)$$

Le membre de droite de (4.10) dépend de ε , on le note $r(\varepsilon)$ et on obtient pour différentes valeurs de ε le tableau suivant

ε	$r(\varepsilon)$	ε	$r(\varepsilon)$
10^{-1}	0,996666666666667	10^{-7}	0,999999999999997
10^{-2}	0,999966666666667	10^{-8}	1,000000000000000
10^{-3}	0,999999666666667	10^{-9}	1,000000000000000
10^{-4}	0,999999996666667	10^{-10}	1,000000000000000
10^{-5}	0,999999999966667	10^{-11}	1,000000000000000
10^{-6}	0,999999999996667	10^{-12}	1,000000000000000

TAB. 4.1 – Méthode de la variable complexe

4.1.2.3 Linéarisation

Pour réduire l'influence de ε , une approche utilisée dans le calcul variationnel est de calculer avec les différences finies uniquement les dérivées partielles. Ainsi, si on note $\delta K = \varepsilon e_i$ et $\delta C = \partial_K C \delta K$ la variation de C , on obtient par linéarisation de $E(K, C(K)) = 0$

$$\frac{\partial E}{\partial C} \delta C = -\frac{\partial E}{\partial K} \delta K \approx -[E(K + \varepsilon K_i, C(K + \varepsilon K_i)) - E(K, C(K))] \quad (4.11)$$

On calcule ensuite la variation totale du gradient de J par rapport à K

$$\frac{dJ}{dK} = \frac{\partial J}{\partial K} + \frac{\partial J}{\partial C} \frac{\partial C}{\partial K} \quad (4.12)$$

où les dérivées partielles sont calculées par différences finies sauf $\frac{\partial C}{\partial K}$ qui est obtenue en résolvant (4.11). Cependant, comme pour les deux méthodes précédentes, le coût du calcul reste toutefois proportionnel à la taille du vecteur K .

4.1.2.4 Méthode de l'état adjoint

On a vu dans le chapitre 3 que la méthode de l'état adjoint permettait, en résolvant seulement deux systèmes d'équations aux dérivées partielles, d'obtenir le gradient de la fonction coût. Si on prend comme modèle (3.1) avec F défini par (4.2), alors le gradient est donné par la formule (3.17)

$$\frac{\partial J}{\partial K} = - \int_0^T \left(\frac{\partial F}{\partial K} \right)^* P dt.$$

On voit que le coût est proportionnel à la taille de la variable d'état et à la taille du paramètre de contrôle et qu'il reste à évaluer le terme $\frac{\partial F}{\partial K}$ (par exemple par différences finies ou par différentiation automatique).

En pratique, la bonne approche consiste à discrétiser l'équation d'état, l'implémenter dans un programme informatique (c'est le code `Traces`) et ensuite différencier ce code (et donc les équations discrètes) en mode inverse par différentiation automatique. Cette méthode est étroitement liée à l'état adjoint ([52, 57, 92])* et nous permet de calculer le gradient de façon exacte. C'est cette technique que nous utiliserons par la suite.

On aurait aussi pu, en vue d'obtenir le code adjoint, définir un **Lagrangien** discret et les équations adjointes discrètes associées aux équations discrètes du code direct et ainsi calculer un gradient discret qui aurait été exact. Cependant, cette méthode est très longue à mettre en place et toute modification du code source entraîne une modification du code adjoint. C'est pour cette raison que nous avons choisi d'automatiser la procédure d'obtention du code adjoint.

4.2 Différentiation automatique du code par Tapenade

4.2.1 Généralités sur la différentiation automatique

Définition 4.1 *On appelle variables dépendantes les variables de sorties dont on veut les dérivées.*

Définition 4.2 *On appelle variables indépendantes les variables d'entrées par rapport auxquelles les variables de sorties seront différenciées.*

Définition 4.3 *On appelle variables actives des variables dépendantes ou indépendantes.*

*Ces trois références nous donnent les grands principes de la différentiation automatique (définitions, temps mémoire, coût, etc...) et comparent différents outils de différentiation.

Pour dériver tout ou partie d'un code, il convient d'abord d'identifier les **variables actives**, qui sont celles par rapport auxquelles on va dériver, et les variables passives qui y interviennent. Les variables actives sont tout simplement celles qui sont modifiées. Par exemple, les variables d'état C (la concentration) et h (la charge) sont actives. A chaque variable active ϕ (appelée variable directe) on associe une variable tangente $\delta\phi$.

4.2.1.1 Principe de la dérivation automatique

Dans ce paragraphe, on explique la façon de procéder sur un exemple pour générer le code tangent et le code adjoint. On considère la ligne de code suivante :

$$x = ay + bz^2 + cf(t) + d \quad (4.13)$$

où x , y , z et t sont les variables actives, a , b , c et d sont des variables passives et f est une fonction au moins C^1 . Le **code tangent** doit permettre de calculer δx au premier ordre, lorsque toutes les variables actives ϕ sont perturbées en $\phi + \delta\phi$. On commence donc par perturber la ligne (4.13) :

$$\begin{aligned} x + \delta x &= a(y + \delta y) + b(z + \delta z)^2 + cf(t + \delta t) + d \\ &= a(y + \delta y) + b(z^2 + 2z\delta z + (\delta z)^2) + c(f(t) + \delta t f'(t) + o(\delta t)) + d. \end{aligned} \quad (4.14)$$

Puis on ne garde que les termes d'ordre au plus 1, et enfin on soustrait x pour avoir :

$$\delta x = a\delta y + 2bz\delta z + cf'(t)\delta t. \quad (4.15)$$

Le **code tangent** est le code d'évolution des variables tangentes et on peut remarquer qu'il fait intervenir les variables directes, qui devront donc être stockées, dès que celles-ci interviennent de manière non-linéaire. Finalement, pour une ligne de code faisant intervenir des fonctions aux moins C^1 , dériver un code revient à un simple calcul de différentielle. Les difficultés commencent lorsque les fonctions qui interviennent ne sont plus dérivables.

Le **code adjoint** est obtenu par transposition du code tangent. Comme précédemment, nous ne rentrons pas dans les détails et nous nous contenterons d'illustrer l'idée générale sur un exemple. Pour transposer la ligne (4.15), on l'écrit sous forme matricielle :

$$\begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta t \end{pmatrix} = \begin{pmatrix} 0 & a & 2bz & cf'(t) \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \\ \delta z \\ \delta t \end{pmatrix} \quad (4.16)$$

Puis on transpose (4.16)

$$\begin{pmatrix} x^* \\ y^* \\ z^* \\ t^* \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ a & 1 & 0 & 0 \\ 2bz & 0 & 1 & 0 \\ cf'(t) & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x^* \\ y^* \\ z^* \\ t^* \end{pmatrix} \quad (4.17)$$

où ϕ^* est appelé variable adjointe de ϕ . Enfin on revient au code en ligne, en exécutant les instructions (4.17) dans l'ordre inverse, de la dernière ligne à la première, *i.e.*

$$\begin{cases} t^* &= t^* + cf'(t)x^* \\ z^* &= z^* + 2bz x^* \\ y^* &= y^* + ax^* \\ x^* &= 0 \end{cases} \quad (4.18)$$

Les quatre lignes (4.18) sont appelées code adjoint de la ligne (4.13).

On peut résumer l'obtention du code adjoint en trois étapes

1. dériver les instructions,
2. inverser l'ordre des instructions,
3. transposer chaque instruction.

4.2.1.2 Difficultés

Les principales difficultés résident dans la non dérivabilité (éventuelle) de certaines fonctions et la gestion de la mémoire du code adjoint. Nous venons de voir dans (4.15) et (4.18) que lorsque le code direct n'est pas linéaire, les valeurs des variables directes sont requises dans les codes tangent et adjoint. Pour obtenir ces valeurs au fur et à mesure de l'exécution du code tangent (puis du code adjoint) il existe plusieurs possibilités :

1. stocker toutes les valeurs en mémoire, ce qui est possible si les vecteurs ne sont pas de trop grande taille ;
2. écrire toutes les valeurs dans des fichiers sur le disque de la machine, et les consulter au fur et à mesure en accédant au disque, ce qui augmente le temps de calcul ;
3. recalculer les valeurs au fur et à mesure, ce qui peut être aussi extrêmement coûteux ;
4. écrire une partie des valeurs sur le disque, les consulter au fur et à mesure et recalculer les valeurs intermédiaires non stockées ;
5. écrire une partie des valeurs sur le disque, les consulter au fur et à mesure et les interpoler pour approcher les valeurs intermédiaires non stockées.

Le choix de la méthode dépend bien sûr de la taille du problème et de ses particularités, notamment son caractère faiblement ou fortement non linéaire. Il faut garder à l'esprit que les accès au disque prennent beaucoup de temps et que les points 1 et 5 doivent être préférés dès que possible. En ce qui concerne Tapenade, il privilégie l'approche 2, tout en proposant des évolutions vers l'approche intermédiaire 4.

4.2.2 Présentation de Tapenade

Tapenade est le successeur d'Odyssée qui fût développé par l'équipe Safir de l'INRIA Sophia-Antipolis. L'outil Tapenade, développé par l'équipe Tropics à l'INRIA [92], calcule les dérivées directionnelles (mode tangent) et les gradients d'un code écrit en fortran 77 ou en fortran 90/95.

La différentiation automatique ([52, 57, 59]) est une technique pour évaluer les dérivées d'une fonction définie par un programme informatique. La différentiation automatique, comme les différences divisées, demande seulement le programme original P . Mais à la place d'exécuter P sur plusieurs jeux d'entrées, le logiciel de différentiation construit un nouveau programme augmenté P' , qui calcule les dérivées analytiques le long du programme original. Ce nouveau programme est appelé le programme différentié. Chaque fois

que le programme original prend la valeur X en entrée, le programme différentié prend une valeur supplémentaire dX , la différentielle de X .

$$\frac{P(X + \varepsilon dX) - P(X)}{\varepsilon} \simeq P'(X).dX$$

Tapenade est basée sur la transformation de source*, qui consiste à ajouter au programme les nouvelles variables, tableaux, et structures de données qui contiendront les dérivées, et à ajouter les nouvelles instructions qui calculent ces dérivées. Le programme qui en résulte peut être compilé en un code efficace. D'autres logiciels, comme ADOL-C [1], sont basés sur la surcharge d'opérateurs qui consiste à dire au compilateur que chaque nombre réel est remplacé par une paire de nombres réels, le deuxième prenant la différentielle. Pour plus d'informations sur la manière dont sont calculées les dérivées, on renvoie au site de l'équipe Tropics <http://www-sop.inria.fr/tropics/>.

Tapenade peut fonctionner en mode tangent (dérivées directionnelles) et en mode inverse (gradients). Nous allons détailler ces deux modes dans les paragraphes suivants.

4.2.2.1 Fonctionnalités de Tapenade

En différentiation automatique, on identifie les programmes avec des séquences d'instructions, elles-mêmes identifiées avec des fonctions composées. Précisément, la séquence d'instructions

$$I_1; I_2; \dots; I_{p-1}; I_p;$$

est identifiée à la fonction :

$$f = f_p \circ f_{p-1} \circ \dots \circ f_1.$$

Le code direct peut donc être représenté par une séquence d'instructions. Si on prend K comme vecteur d'entrée, le code de simulation Traces est identifié à

$$K \xrightarrow{f_1} f_1(K) \xrightarrow{f_2} f_2 \circ f_1(K) \xrightarrow{f_3} \dots \xrightarrow{f_p} f(K) = f_p \circ \dots \circ f_1(K) \xrightarrow{\tilde{J}} c$$

où $c \in \mathbb{R}$ est la valeur de la fonction coût $J(K)$.

★★ Mode tangent

En mode tangent, le programme différentié est juste une copie du programme original, les instructions contenant les dérivées sont insérées avant chaque instruction qui implique des variables actives.

Pour une sortie $c = J(K) = \tilde{J} \circ f_p \circ f_{p-1} \circ \dots \circ f_1(K)$ du code direct, le code tangent retourne (formellement)

$$\delta c = \tilde{J}'(f(K)) \cdot f_p'(K_{p-1}) \cdot f_{p-1}'(K_{p-2}) \dots f_1'(K_0) \delta K \quad (4.19)$$

où les K_n , $n = 1, \dots, p-1$, sont les valeurs des variables juste après avoir exécuté les n premières fonctions (on pose $K_0 = K$) et les f_n' , $n = 1, \dots, p-1$, sont les matrices Jacobiennes.

Le code tangent doit permettre de calculer la variation coût δc associé à une variation δK du paramètre. On a la relation suivante liant le gradient à la variation δK

$$\delta c = (\nabla J(K), \delta K). \quad (4.20)$$

*On peut également citer le compilateur TAMC [91] pour le langage fortran.

Donc si on prend $\delta K = e_i$ où e_i est le $i^{\text{ème}}$ vecteur de base, alors la formule (4.20) nous donne la $i^{\text{ème}}$ composante du gradient. Cette méthode nécessite N intégrations (la longueur de K) du code tangent et une intégration du code direct.

★★ Mode inverse

En mode inverse, le programme généré par Tapenade calcule le gradient de la fonction coût. Partant du code tangent (4.19), on obtient que

$$\delta c = \tilde{J}'(f(K)) \cdot f'_p(K_{p-1}) \cdot f'_{p-1}(K_{p-2}) \dots f'_1(K_0) \delta K \quad (4.21)$$

$$= (\tilde{J}'(f(K)) \cdot f'_p(K_{p-1}) \cdot f'_{p-1}(K_{p-2}) \dots f'_1(K_0) \delta K, 1)_{\mathbb{R}} \quad (4.22)$$

$$= (f'_p(K_{p-1}) \cdot f'_{p-1}(K_{p-2}) \dots f'_1(K_0) \delta K, \tilde{J}'(f(K))^T 1)_{\mathbb{R}} \quad (4.23)$$

$$\begin{aligned} & \vdots \\ & = (\delta K, [f'_1(K_0)]^T \dots [f'_p(K_{p-1})]^T [\tilde{J}'(f(K))]^T 1)_{\mathbb{R}}. \end{aligned} \quad (4.24)$$

Finalement avec (4.20), on arrive à

$$\nabla J(K) = [f'_1(K_0)]^T \dots [f'_p(K_{p-1})]^T [\tilde{J}'(f(K))]^T 1. \quad (4.25)$$

Cependant, on observe que les K_n sont requis dans l'ordre inverse du calcul. Si le programme original réécrit une partie des K_n , le programme différentié doit les restaurer avant qu'ils soient utilisés par $f'_{n+1}(K_n)$.

4.2.2.2 Checkpointing

Tapenade utilise la stratégie “Store All” : les K_n sont restaurés à partir d'une pile en cas de besoin. Cette pile est remplie au cours d'une exécution préliminaire du programme, appelé le “balayage avant”, qui stocke les variables additionnelles dans la pile juste avant qu'elles ne soient réécrites. On peut voir sur la figure 4.2 qu'avant chaque instruction du balayage avant, des valeurs sont sauveées (points noirs) et qu'elles sont restaurées (points blancs) juste avant leurs instructions dérivées correspondantes pendant le “balayage arrière”.

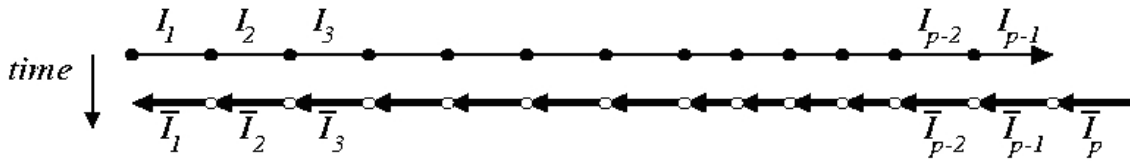


FIG. 4.2 – Stratégie “Store All”

Cette stratégie a besoin d'un compromis particulier, appelé **checkpointing**, entre le stockage et le “recalcul” pour être vraiment rentable. Le checkpointing consiste à sélectionner une partie du programme, et à ne pas stocker les valeurs écrasées par cette partie, au prix d'une mémorisation supplémentaire de l'état complet du programme, que nous appelons une “photo”. Le checkpointing semble être la meilleure solution pour gérer la mémoire.

Le résultat du checkpointing sur la partie P avec le mode Store All est montré sur le haut de la figure 4.3. La photo (gros point noir) mémorise l'état du programme juste avant P ce

qui nécessite de dupliquer l'exécution de P et engendre donc un coût supplémentaire. La partie du bas de la figure 4.3 montre l'exécution du programme quand il y a de nombreux points de contrôle imbriqués : moins de valeurs sont mémorisées, mais il en coûte des photos et des exécutions supplémentaires.

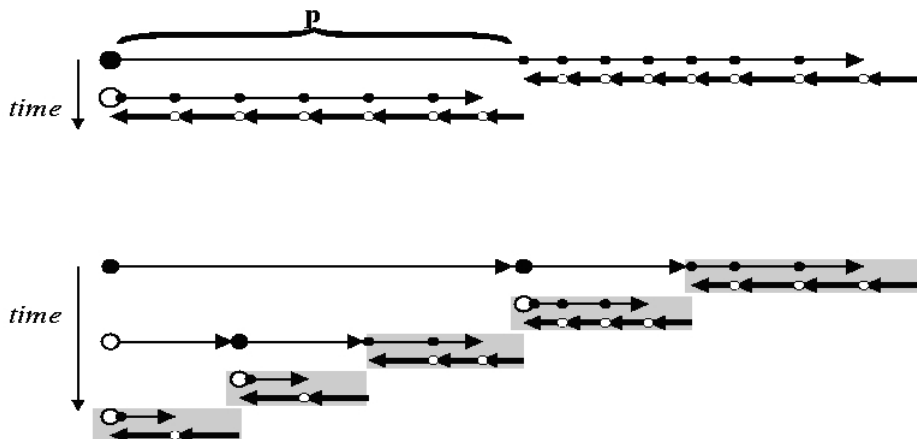


FIG. 4.3 – Checkpointing

4.2.2.3 Utilisation de Tapenade

Il faut préciser à Tapenade plusieurs choses avant de différentier le code :

- la routine dans laquelle se trouve la fonction à dériver (dans notre cas c'est `traces0`) ;
- les variables d'entrées, qui dans notre cas sont la porosité et le coefficient de diffusion, et la variable de sortie, *i.e.* la valeur de la fonction coût (la variable par rapport à laquelle on va dériver) ;
- le mode (tangent ou adjoint).

Après les différentes étapes décrites ci-dessus, on lance Tapenade qui analyse tout d'abord les routines (avec un parser fortran) ce qui lui permet d'avoir l'architecture du code ainsi que les dépendances entre les routines et les variables. A la suite de ça, Tapenade garde les routines dont il a besoin et commence la différentiation du code à proprement dit.

Remarque 4.1 *Tapenade rajoute la lettre “d” aux variables différentiées en mode tangent et la lettre “b” en mode adjoint ainsi que le terme “_d” aux noms des routines différentiées en mode tangent et le terme “_b” en mode adjoint.*

4.2.2.4 Code adjoint généré par Tapenade

Si on reprend la structure du code `Traces` (cf. section 2.4.1), on a comme architecture du code adjoint de `Traces` un diagramme similaire au diagramme 2.2.

Remarque 4.2 *Les routines `traces` et `calcul traces` n'apparaissent pas dans l'arborescence du code adjoint de `Traces` car elles ne sont pas différentiées. En effet, après analyse du code, Tapenade n'a trouvé dans ces routines aucune ligne de code à dériver, il ne les a donc pas prises en compte lors de la différentiation.*



FIG. 4.4 – Architecture du code adjoint de Traces

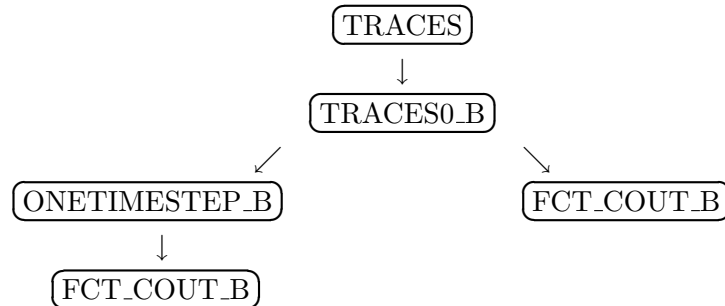


FIG. 4.5 – Structure du fichier traces0_b.f90

Gradients

Pour générer le code adjoint, on lance Tapenade en mode inverse : celui-ci nous renvoie les routines différenciées, la routine de tête étant `traces0_b` (figure 4.4). Le gradient de la fonction coût est alors accessible dans la variable `costb` de la routine `fct_cout_b` (cf. section 2.4.1 et remarque 4.1). Une fois le gradient de la fonction coût calculé, il ne reste plus qu'à le transmettre à l'optimiseur pour commencer la minimisation.

Le tableau 4.2 donne les grandes lignes de l'algorithme de résolution numérique du problème inverse qui a été mis en place au chapitre 3.

Algorithme 4.1 Résolution numérique du problème inverse

- 1: **Données** (entrées du code) : vecteur $x_0 := (\omega, D)$
 - 2: **while** $\nabla J(x_k) \neq 0$ **do**
 - 3: **call** traces0
 - 4: $J(x_k) = \text{cost}$
 - 5: **call** traces0_b
 - 6: $\nabla J(x_k) = \text{costb}$
 - 7: Tester la convergence
 - si le minimum est atteint, **stop**
 - sinon $x_{k+1} = x_k - \alpha_k B_k^{-1} \nabla J(x_k)$ et retour à l'étape 3
 - 8: **end do**
-

TAB. 4.2 – Algorithme de résolution numérique du problème inverse

4.3 Du point de vue numérique

4.3.1 Validation des codes générés automatiquement

Afin de valider le code adjoint, *i.e.* vérifier que le gradient obtenu avec le code adjoint généré par Tapenade est juste, on fait généralement le “test du gradient”. C’est une façon de comparer la valeur du gradient calculée par le code adjoint à celle obtenue par les différences divisées. Si on prend notre vecteur des paramètres $K \in \mathbb{R}^N$ et $\delta K \in \mathbb{R}^N$ avec $\|\delta K\| = 1$, on a

$$J(K + \delta K) - J(K) = \langle \nabla J(K), \delta K \rangle + o(\|\delta K\|). \quad (4.26)$$

Si pour toute direction de perturbation δK , on vérifie

$$r(\varepsilon) = \frac{J(K + \varepsilon \delta K) - J(K)}{\varepsilon \langle \nabla J(K), \delta K \rangle} \xrightarrow{\varepsilon \rightarrow 0} 1 \quad (4.27)$$

alors le test du gradient est satisfait. Ce test permet de savoir si le code adjoint fournit un calcul exact du gradient. En regardant les valeurs calculées (pour une direction donnée) dans le tableau 4.3, on peut être confiant sur la valeur du gradient obtenue avec le générateur automatique de code Tapenade. On voit en effet que pour les valeurs de ε comprise entre 10^{-7} et 10^{-16} , le rapport $r(\varepsilon)$ est quasiment égal à 1. A partir de 10^{-17} , on est lié à la précision machine, il n’est donc pas étonnant d’obtenir des valeurs aberrantes pour $\varepsilon \leq 10^{-17}$.

Il est très **important** de faire ce test avant d’utiliser un algorithme de minimisation car transmettre un gradient faux ou imprécis à l’optimiseur ne permet pas de converger vers la solution du problème. De plus, les sources d’erreurs provenant de la résolution numérique des problèmes inverses peuvent être assez nombreuses (cf. section 5.1), il est donc nécessaire d’être sûr des valeurs numériques du gradient.

ε	$r(\varepsilon)$	ε	$r(\varepsilon)$
10^{-5}	0,1639988539827272	10^{-14}	1,0001224546473169
10^{-6}	0,7476895636206117	10^{-15}	0,9993793587795462
10^{-7}	0,9668650856806020	10^{-16}	1,0148354227477239
10^{-8}	0,9965739403864012	10^{-17}	1,1169131116352610
10^{-9}	0,9996562197586708	10^{-18}	0,4538619958621305
10^{-10}	0,9999656160742779	10^{-19}	9,4485815502207180
10^{-11}	0,9999965826243937	10^{-20}	2,8882127009408310
10^{-12}	1,0000010094292604	10^{-21}	16,504072576804750
10^{-13}	0,9999999531686156	10^{-22}	0,0

TAB. 4.3 – Précision du test du gradient

4.4 Conclusion

4.4.1 Résumé

Dans ce dernier paragraphe, nous résumons l'étude qui a été faite sur l'estimation de paramètres (chapitre 3) par un diagramme (figure 4.6) qui retrace le cheminement suivi.

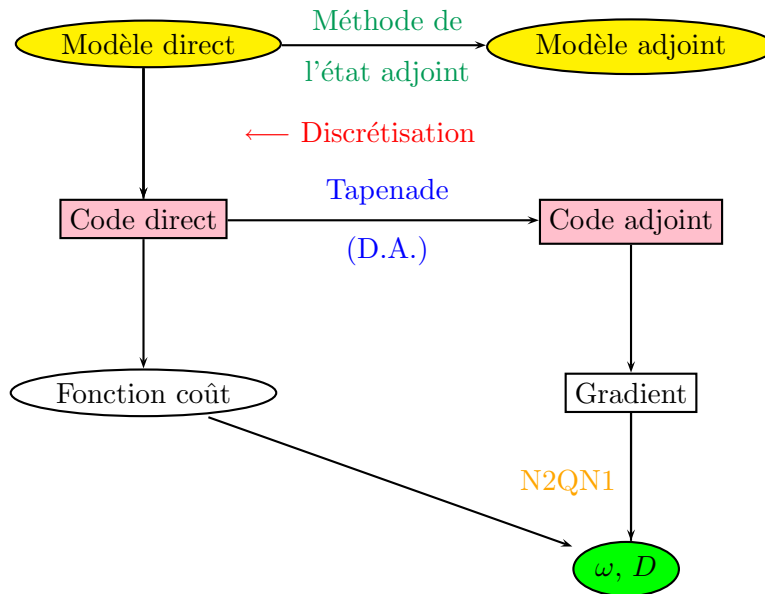


FIG. 4.6 – Diagramme de résolution du problème inverse

4.4.2 Discussion sur la différentiation automatique

Tapenade est un outil efficace pour obtenir des dérivées directionnelles ou des gradients car il permet la production automatique des dérivées et des instructions adjointes. En outre, la différentiation automatique peut être un outil très intéressant pour vérifier une différentiation manuelle. Cependant, les codes produits par différentiation automatique n'utilisent généralement pas la mémoire de l'ordinateur de manière très efficace. La sauvegarde du modèle direct est le problème majeur. Il faut également préciser que dans notre cas la différentiation automatique a bien fonctionné mais il y a eu beaucoup de travail pour arriver à obtenir un code adjoint correct ainsi qu'un gradient juste (cf. chapitre 1 section 1.4.3, chapitre 2 section 2.4.1, chapitre 3 section 3.3.5).

Actuellement, le code tangent prend (en moyenne) environ 3 fois le temps de calcul du code direct et le code adjoint 7 fois. Une future étape de notre travail consisterait donc à optimiser la méthode de checkpointing en utilisant l'algorithme binomial de Griewank & Walther [55] qui nous donnerait un compromis optimal entre le temps et la consommation mémoire.

Chapitre 5

Estimation de paramètres

Sommaire

5.1	Généralités	72
5.2	Résultats numériques	72
5.2.1	Cas-test “colis”	72
5.2.1.1	Données non bruitées	73
5.2.1.2	Données bruitées	75
5.2.2	Cas-test “réel”	77
5.2.2.1	Données non bruitées	77
5.2.2.2	Données bruitées	81
5.3	Conclusion des tests	90

Afin de tester la validité de la méthode présentée dans le chapitre 3 et de voir les possibilités et les limites de l’algorithme, nous avons effectué un grand nombre de simulations que nous présentons dans ce chapitre. Après un paragraphe introductif, nous exposons les résultats obtenus pour les deux cas-tests définis au chapitre 2.

5.1 Généralités

Nous avons généré les données de concentration avec le code direct. On parlera ainsi de données synthétiques pour les distinguer des données réelles issues de mesures expérimentales (comme pour l’expérience DIR). C’est à partir de ces données de concentration qu’on essaye de déterminer la diffusion et la porosité. Comme nous l’avons mentionné dans le chapitre 1, l’avantage d’utiliser des données synthétiques est que l’on peut comparer la porosité et le champ de diffusion obtenus en résolvant le problème inverse avec la porosité et le champ de diffusion initiaux (qui sont en pratique inconnus). Ce travail est absolument indispensable puisque dans un premier temps il va nous servir à valider les méthodes employées et dans un second temps les équipes de l’Andra l’utiliseront pour l’inversion sur les données réelles issues du laboratoire souterrain de Bure.

La première chose à faire, avant de commencer les tests, est de vérifier le calcul du gradient obtenu par différentiation automatique en comparant sa valeur avec celle calculée par différences finies. Nous avons obtenu des valeurs très satisfaisantes pour la validation du gradient et nous renvoyons à la section 4.3.1 du chapitre 4 pour les résultats. Cette validation est importante pour plusieurs raisons, la principale étant que si l’on transmet un gradient imprécis à l’optimiseur, l’algorithme risque de ne pas converger vers la solution du problème. De plus, l’algorithme peut “tomber” dans un minimum local et ne plus en sortir, le problème direct peut être mal formulé, les données peuvent être en quantité insuffisantes pour identifier les paramètres, etc. Pour toutes ces raisons, il est important de valider le gradient afin de mieux cerner les problèmes rencontrés.

On teste dans cette section les performances de l’algorithme d’inversion dans différentes simulations numériques. Nous traiterons pour chacun des deux cas-tests le cas de données synthétiques non bruitées puis le cas de données synthétiques bruitées.

5.2 Résultats numériques

5.2.1 Cas-test “colis”

On rappelle que le domaine considéré mesure 60 m de longueur par 40 m de largeur et qu’il est constitué de deux zones dont l’une (zone bleue) contient le colis (le radionucléide). Les valeurs des paramètres (constants par zones) sont données par la dernière colonne du tableau 5.1. La diffusion est exprimée en m^2/an et la porosité est sans unité. La fonction coût est exprimée par la formule (3.21). Les poids α et β ont été fixés à 1.

La résolution du problème direct permet d’obtenir les valeurs de concentration à tous les pas de temps dans Ω_2 et les valeurs à l’instant final dans Σ , qui a été réduit pour les mesures à une droite située à 20 m du bord supérieur du rectangle (cf figure 2.5). La simulation s’est effectuée sur 1000 ans avec un pas de temps de calcul de 15 ans. Ce sont ces données de concentration que l’algorithme va essayer d’ajuster en modifiant au cours des itérations les paramètres du modèle.

5.2.1.1 Données non bruitées

Le nombre de paramètres à identifier s'élève à quatre : un paramètre de diffusion et de porosité par zone. Les résultats de l'inversion sont exposés dans le tableau 5.1 : la deuxième colonne montre les valeurs initiales données au minimiseur, la troisième colonne présente les paramètres identifiés et la dernière colonne rappelle les valeurs réelles. Le tableau montre que l'identification a parfaitement fonctionné. L'algorithme a nécessité 57 itérations pour converger vers la bonne solution.

La recherche des paramètres de diffusion s'est réalisée entre les valeurs $10^{-12} m^2/an$ (borne inférieure) et $10^{-1} m^2/an$ (borne supérieure) et ceux de porosité entre 0,01 % et 100 %. Leurs évolutions au cours des itérations, ainsi que celle de la fonction coût et du gradient, sont présentées sur la figure 5.1.

Paramètres	initiaux	identifiés	réels
D_1	0.001	$9,4999999999 \cdot 10^{-7}$	$9,5 \cdot 10^{-7}$
D_2	0,001	0,0002000000	0,0002
ω_1	0,1	0,0009999999	0,001
ω_2	0,1	0,3000000000	0,3

TAB. 5.1 – Valeurs identifiées pour le cas-test "colis" sans bruit

Plusieurs autres tests ont été réalisés en partant d'autres données initiales. L'algorithme a également convergé vers la solution exacte entre 40 et 100 itérations suivant les valeurs de départ.

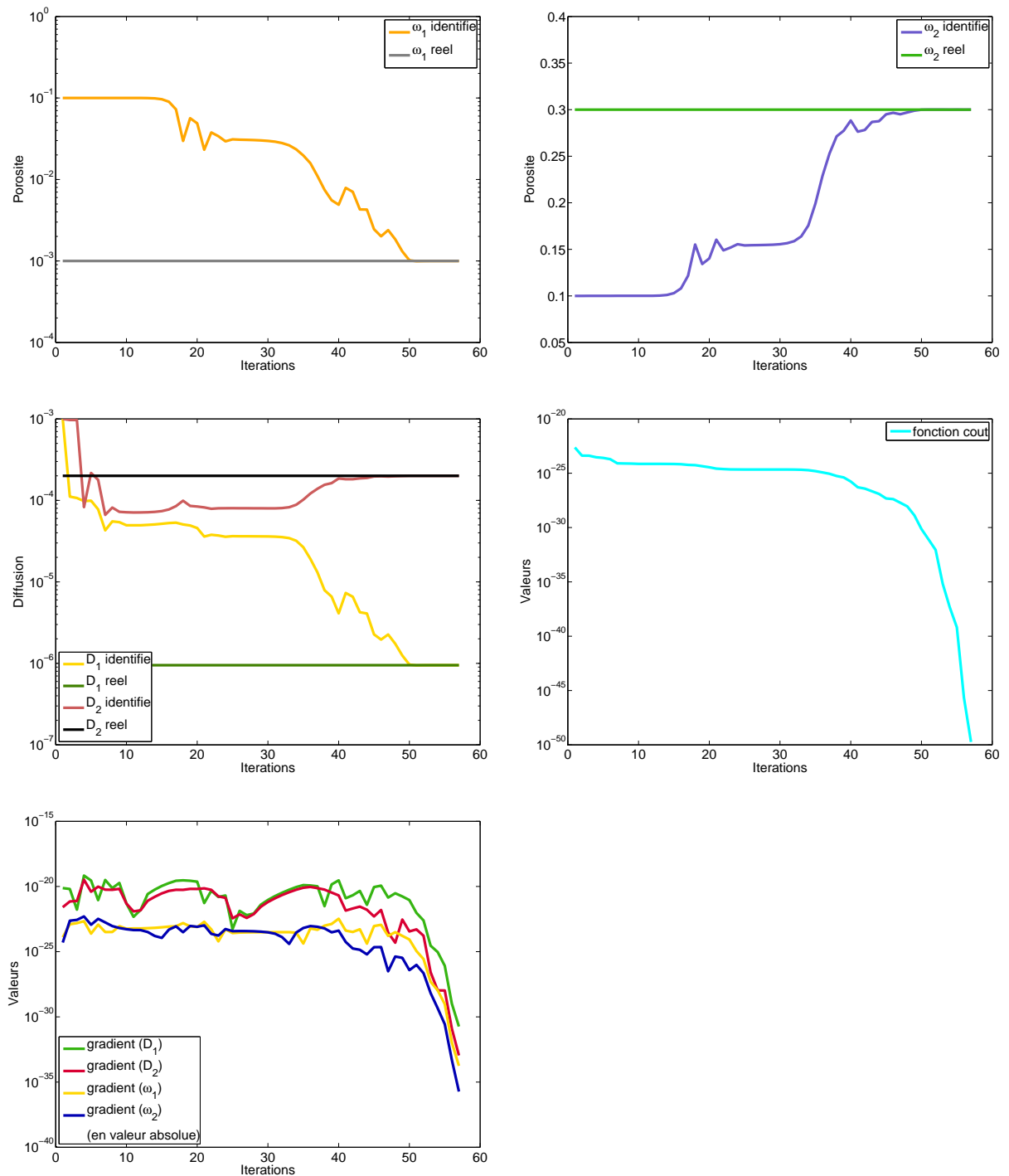


FIG. 5.1 – En haut : évolution des 2 paramètres de porosité en échelle semi-log à gauche. Au milieu à gauche : évolution des 2 paramètres de diffusion en échelle semi-log. Au milieu à droite : évolution de la fonction coût en échelle semi-log. En bas à gauche : évolution des 4 composantes du gradient en échelle semi-log.

Remarque 5.1 *La minimisation s'arrête dans l'un des cas suivants :*

- le nombre maximal d'itérations a été atteint,
- $N2QN1$ est tombé sur l'un de ces critères d'arrêt,
- le pas de descente est devenu trop petit,
- la direction de descente n'en est plus une.

5.2.1.2 Données bruitées

On se donne maintenant des données synthétiques bruitées. La loi de distribution choisie pour la génération du bruit est une loi uniforme, de moyenne nulle, et normalisée à la même taille que les observations. C'est une loi simple qui a l'avantage d'être facile à implémenter. Le bruit est donc au départ normalisé à la même taille que les observations puis on applique un coefficient multiplicatif pour régler le niveau du bruit, proportionnellement aux données, avec un rapport ajustable (dans le test ci-dessous, il a par exemple été fixé à 5%).

La recherche des paramètres de diffusion s'est réalisée entre les valeurs $10^{-12} m^2/an$ (borne inférieure) et $10^{-1} m^2/an$ (borne supérieure) et ceux de porosité entre 0,01 % et 100 %. L'algorithme a tourné un peu plus longtemps (83 itérations) que dans le cas sans bruit (57 itérations) mais il n'a pas réussi à converger complètement vers la solution puisque seuls les paramètres de la zone Ω_2 ont été correctement identifiés.

Il n'est pas étonnant que les paramètres dans Ω_1 (zone la plus grande) soient mal identifiés, on a en effet moins d'informations dans cette zone du à un phénomène de diffusion assez lent et aux mesures **faites** seulement à l'instant final sur une droite éloignée du colis. Mais, comme on le verra dans la suite, le fait de mettre des poids adaptés permettra de rééquilibrer les deux termes de la fonction coût et ainsi d'avoir une meilleure identification des paramètres (cf. section 5.2.2.1). De plus, le bruit a été généré de la même manière dans les trois zones, c'est-à-dire qu'il y a (pour un bruit de 5%) une erreur sur la concentration comprise entre 0 et 0,05. Donc proportionnellement, il y a très peu de bruit dans le colis (0.05 sur des valeurs de concentration de l'ordre de 1, c'est très faible) et beaucoup plus loin du colis (là où la concentration est quasi nulle), il y a un bruit très grand.

Paramètres	initiaux	identifiés	réels
D_1	0,001	$1,0000000000 \cdot 10^{-12}$	$9,5 \cdot 10^{-7}$
D_2	0,001	0,0002025143	0,0002
ω_1	0,1	$1,0225269332 \cdot 10^{-9}$	0,001
ω_2	0,1	0,3042499904	0,3

TAB. 5.2 – Valeurs identifiées avec un bruit de 5%

L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.2.

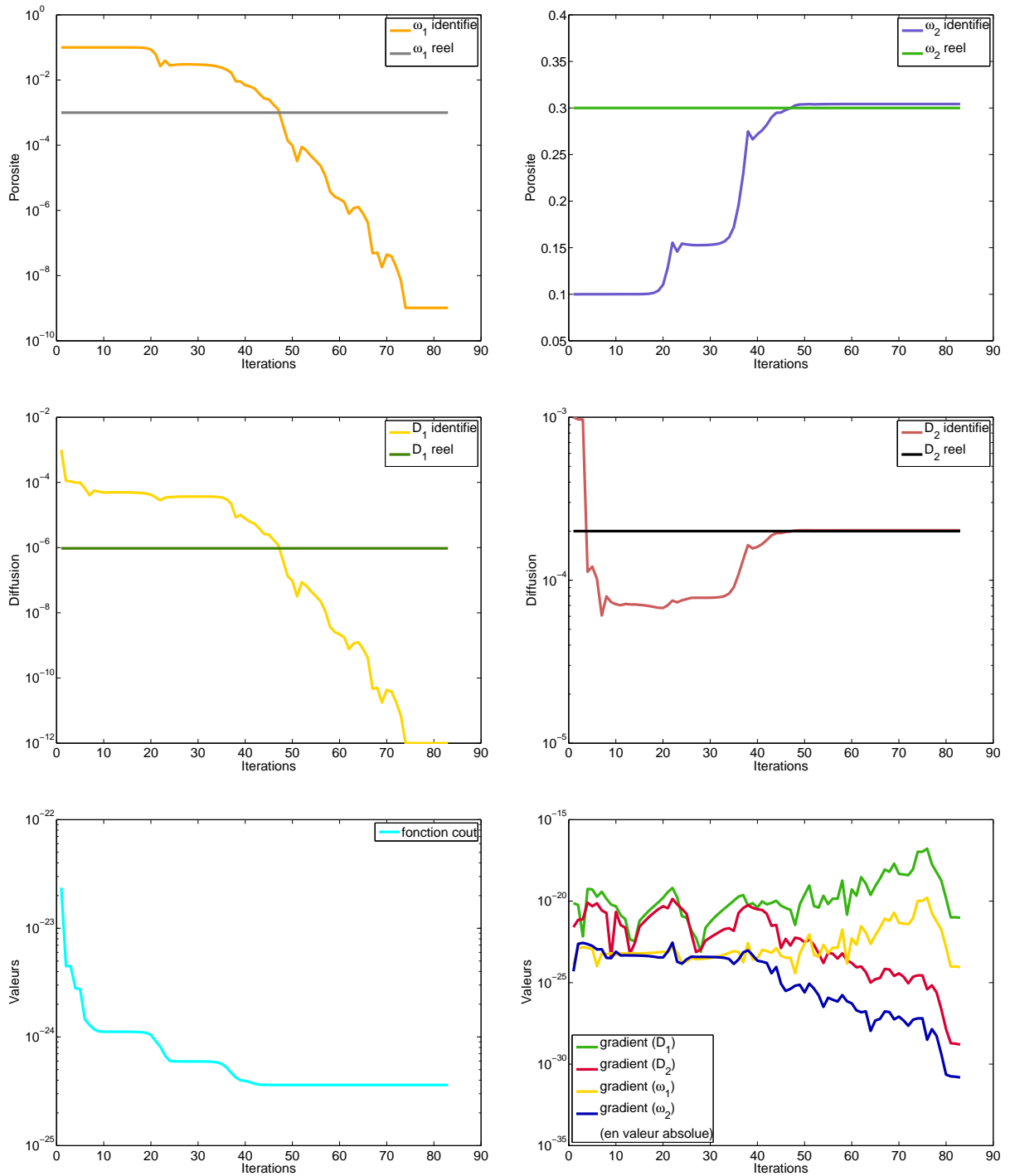


FIG. 5.2 – En haut : évolution des 2 paramètres de porosité en échelle semi-log à gauche. Au milieu : évolution des 2 paramètres de diffusion en échelle semi-log. En bas à gauche : évolution de la fonction coût en échelle semi-log. En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

5.2.2 Cas-test “réel”

Le problème étant symétrique (comme pour le problème direct), nous le résolvons uniquement sur la moitié du domaine. On rappelle que le domaine mesure 0,4 m de longueur par 2 m de largeur et qu’il est constitué de trois zones. La diffusion est exprimée en m^2/s et la porosité est sans unité. La fonction coût est exprimée par la formule (3.43).

La résolution du problème direct permet d’obtenir les valeurs de concentration à tous les pas de temps dans Ω_2 et les valeurs à l’instant final dans les domaines $\gamma_i, i = 1, \dots, 3$, qui ont été réduits pour les mesures à deux droites horizontales d’équation $y = 1$ et $y = 0,9$ et une droite verticale d’équation $x = 0$ (cf figure 2.6). La simulation s’est effectuée sur 1000 jours avec un pas de temps de calcul d’une demi-journée. Ce sont ces données de concentration que l’algorithme va essayer d’ajuster en modifiant au cours des itérations les paramètres du modèle.

5.2.2.1 Données non bruitées

Cas non “adimensionné”

Nous essayons dans un premier temps d’identifier les six paramètres : un paramètre de diffusion et de porosité par zone. Les poids α et β ont été fixés à 1. La recherche des paramètres de diffusion s’est réalisée entre les valeurs $10^{-12} m^2/an$ (borne inférieure) et $10^{-6} m^2/an$ (borne supérieure) et ceux de porosité entre 10 % et 2000 %. L’algorithme de minimisation a tourné pendant 36 itérations sans arriver à converger. Les valeurs obtenues sont données dans le tableau 5.3.

Paramètres	initiaux	identifiés	réels
D_1	1, 1.10^{-11}	9,7113117670. 10^{-12}	1, 0.10^{-11}
D_2	1, 1.10^{-6}	1,1000000000. 10^{-6}	1, 0.10^{-6}
D_3	9, 0.10^{-11}	1,6260743312. 10^{-10}	1, 0.10^{-10}
ω_1	0,21	0,2100000000	0,2
ω_2	11,0	11,000000000	10
ω_3	0,29	0,2900000000	0,3

TAB. 5.3 – Identification des 6 paramètres

La dernière colonne du tableau montre que la diffusion n’a pas été correctement identifiée et que la porosité n’a pas bougé de sa valeur initiale. D’autres tests ont été réalisés en partant de données initiales différentes. L’algorithme n’a pas réussi à retrouver les paramètres réels.

Remarque 5.2 *Dans de nombreuses simulations, le minimiseur a souvent commencé par ajuster les valeurs de diffusion avant celles de porosité.*

Ce test n’ayant pas été concluant, nous avons décidé de prendre pour valeurs initiales les valeurs réelles des paramètres dans les zones Ω_2 (zone d’injection) et Ω_3 (zone endommagée). Les résultats sont exposés dans le tableau ci-dessous. L’algorithme de minimisation n’a effectué que 7 itérations sans arriver à converger : la valeur de diffusion D_1 identifiée n’est pas celle recherchée et la valeur de porosité ω_1 est restée bloquée à sa valeur initiale.

Paramètres	initiaux	identifiés	réels
D_1	$1, 5 \cdot 10^{-11}$	$9, 6624324494 \cdot 10^{-12}$	$1, 0 \cdot 10^{-11}$
D_2	$1, 0 \cdot 10^{-6}$	$9, 9999999999 \cdot 10^{-7}$	$1, 0 \cdot 10^{-6}$
D_3	$1, 0 \cdot 10^{-10}$	$9, 9578195668 \cdot 10^{-11}$	$1, 0 \cdot 10^{-10}$
ω_1	0, 21	0, 2100000000	0, 2
ω_2	10	10, 000000000	10
ω_3	0, 3	0, 3000000000	0, 3

TAB. 5.4 – Identification des 2 paramètres de la zone Ω_1

La différence d'ordre de grandeur entre les coefficients de diffusion et de porosité étant très grande (de 10^5 à 10^{12}), elle peut avoir une influence sur les composantes du gradient de la fonction coût et donc sur le choix de la direction de descente de l'algorithme de minimisation. C'est pourquoi, nous avons essayé d'identifier les coefficients de diffusion et de porosité séparément (en fixant dans les deux cas l'autre paramètre à sa valeur réelle). Nous sommes partis avec les mêmes conditions initiales que dans le premier test (*i.e.* avec des valeurs proches de celles recherchées). Pour la porosité, l'algorithme a convergé en 31 itérations vers la bonne solution (tableau 5.5) et pour la diffusion, l'algorithme a convergé en 28 itérations (tableau 5.6).

Paramètres	initiaux	identifiés	réels
ω_1	0, 21	0, 20000000006250	0, 2
ω_2	11, 0	10, 0000000073096	10
ω_3	0, 29	0, 300000000038500	0, 3

TAB. 5.5 – Identification de la porosité

Paramètres	initiaux	identifiés	réels
D_1	$1, 1 \cdot 10^{-11}$	$1, 0000043918 \cdot 10^{-11}$	$1, 0 \cdot 10^{-11}$
D_2	$1, 1 \cdot 10^{-6}$	$1, 1000000000 \cdot 10^{-6}$	$1, 0 \cdot 10^{-6}$
D_3	$9, 0 \cdot 10^{-11}$	$9, 9998884949 \cdot 10^{-11}$	$1, 0 \cdot 10^{-10}$

TAB. 5.6 – Identification de la diffusion

Cas “adimensionné”

Comme nous l'avons expliqué dans le paragraphe précédent, les ordres de grandeur entre les coefficients de diffusion et de porosité étant très grands, nous ne pouvons pas identifier les six paramètres en même temps. Nous nous sommes également aperçus au cours des tests que nous n'arrivons pas à identifier les paramètres de la zone 2. Ce constat n'est pas rédhibitoire car Ω_2 constitue la zone d'injection du radionucléide, nous connaissons donc les

quantités introduites dans cette région. Pour cette raison, on se préoccupera **seulement des zones 1 et 3** dans la suite.

N'ayant pas beaucoup de paramètres à identifier (au maximum six), nous voudrions pouvoir les déterminer ensemble sans avoir à résoudre deux problèmes de minimisation (un pour la diffusion et un pour la porosité). Dans ce but, nous avons adimensionné les paramètres dans l'équation (1.6). Ainsi, la vitesse de Darcy et le terme source étant nuls, nous avons posé

$$D = 10^{-10} D' \quad (5.1)$$

$$\lambda = 10^{-10} \lambda' \quad (5.2)$$

$$t = 10^{10} t' \quad (5.3)$$

$$\text{avec} \quad D' = \begin{cases} 10^{-1} & \text{dans } \Omega_1 \\ 10^4 & \text{dans } \Omega_2 \\ 1 & \text{dans } \Omega_3 \end{cases} \quad (5.4) \quad \omega = \begin{cases} 0,2 & \text{dans } \Omega_1 \\ 10 & \text{dans } \Omega_2 \\ 0,3 & \text{dans } \Omega_3 \end{cases} \quad (5.5)$$

ce qui nous permet d'avoir les valeurs des quatre paramètres à identifier (D_1 , D_3 , ω_1 et ω_3) proches de l'unité.

Dans toute la suite, on assimilera D à D' et on **fixe** les valeurs dans Ω_2 à 10 pour la porosité et à 10^4 pour la diffusion. On prend comme poids pour la fonction coût (3.43)

$$\alpha = 1 \text{ et } \beta = \frac{40 \times 2000}{103}$$

où 40 désigne le nombre de mailles où on mesure la concentration à tous les instants dans Ω_2 , 103 le nombre de mailles où on mesure la concentration à l'instant final sur les droites γ_i avec $i = 1, \dots, 3$ et 2000 le nombre de pas de temps de la simulation. Ces nouveaux poids permettent d'avoir les deux termes de la fonction coût du même ordre (cf figure 5.3).

A partir de ces nouvelles données, nous relançons une minimisation. L'algorithme a convergé après 82 itérations. Les résultats sont exposés dans le tableau ci-dessous : nous sommes partis de valeurs initiales assez éloignées des valeurs réelles et on voit que l'identification a parfaitement fonctionné.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,0999999999	0,1
D'_3	1.10^{-2}	0,9999999999	1,0
ω_1	5.10^{-3}	0,1999999999	0,2
ω_3	2.10^{-3}	0,3000000000	0,3

TAB. 5.7 – Identification dans les zones 1 & 3

L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.3.

Dans la suite, on ne traite plus que le **cas** dit “**adimensionné**” puisqu'il donne des résultats satisfaisants. Dans toutes les simulations qui suivent, la recherche des paramètres de diffusion s'est réalisée entre les valeurs $10^{-5} m^2/an$ (borne inférieure) et $10^2 m^2/an$ (borne supérieure) et ceux de porosité entre 0,01 % et 1000 %.

Remarque 5.3 Grâce à l'adimensionnement des paramètres et au choix adapté des poids de la fonction coût, on a obtenu de meilleurs résultats.

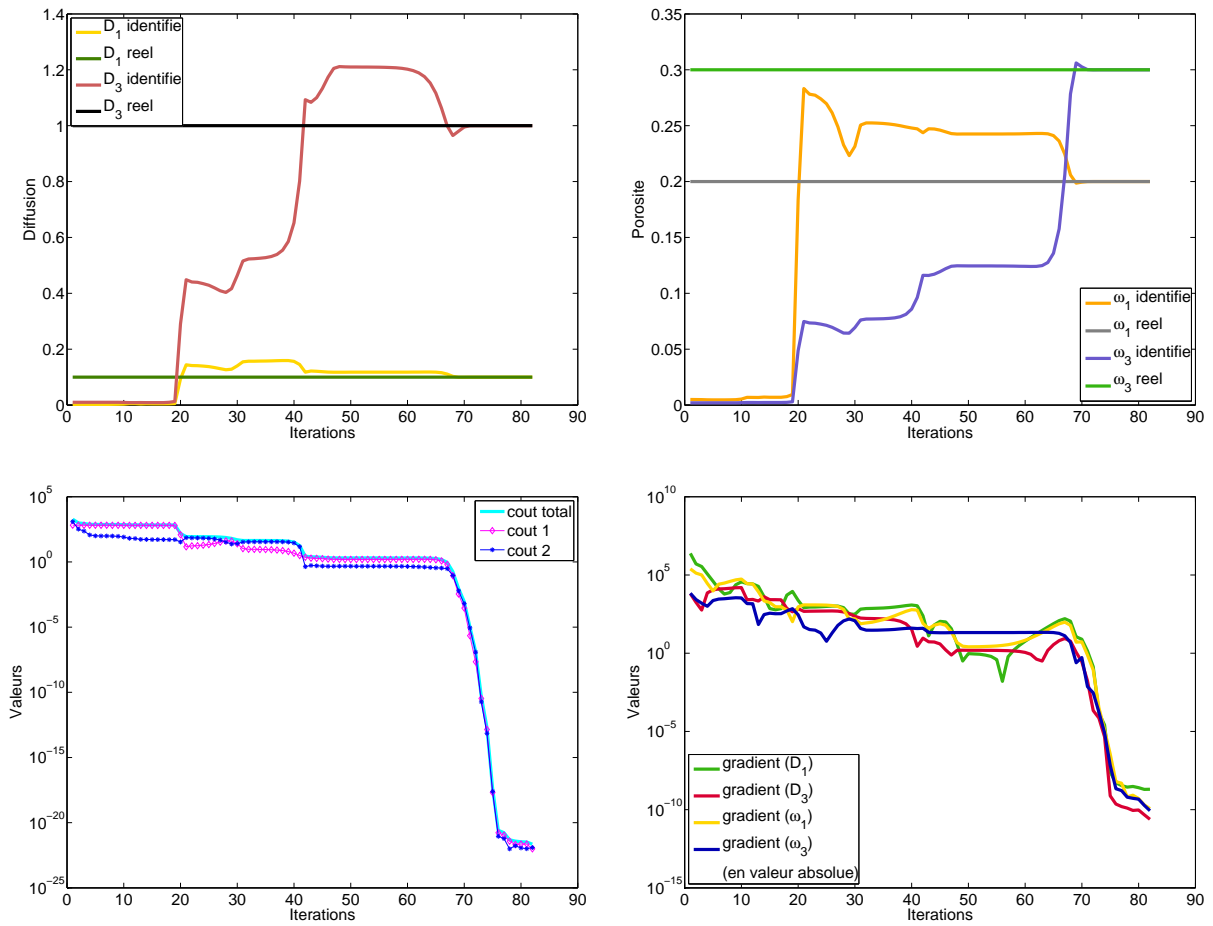


FIG. 5.3 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

5.2.2.2 Données bruitées

Définition 5.1 *La solution de l'inversion est stable si elle est continue par rapport aux données, i.e. une petite erreur sur les données entraîne une petite erreur sur la solution.*

Bruit absolu

Dans cette section, on utilise le même bruit que pour le cas-test “colis” i.e. suivant une loi uniforme, de moyenne nulle, et normalisée à la même taille que les observations.

Dans le cas d'un bruit de 3%, l'algorithme a convergé en 79 itérations et les résultats (tableau 5.8) montrent que la solution est bien approchée dans le cas d'un bruit de faible amplitude. L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.4.

En regardant le tableau ci-dessous, on remarque que le coefficient de porosité ω_3 est moins bien identifié que les autres. La zone endommagée Ω_3 est très mal observée donc dès qu'il y a du bruit ce paramètre est plus difficilement identifiable.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,0991029048	0,1
D'_3	2.10^{-2}	0,9411616874	1,0
ω_1	5.10^{-3}	0,1919263062	0,2
ω_3	2.10^{-3}	0,2349933192	0,3

TAB. 5.8 – Identification dans les zones 1 & 3 avec un bruit de 3%

Remarque 5.4 *Dans tous les tests qui suivent, seule la valeur de ω_3 n'a pas bien été identifiée.*

Remarque 5.5 *Comme pour le cas-test “colis”, le bruit a été généré de la même manière dans les trois zones, i.e. proportionnellement il y a toujours très peu de bruit dans le colis et un bruit élevé loin du colis.*

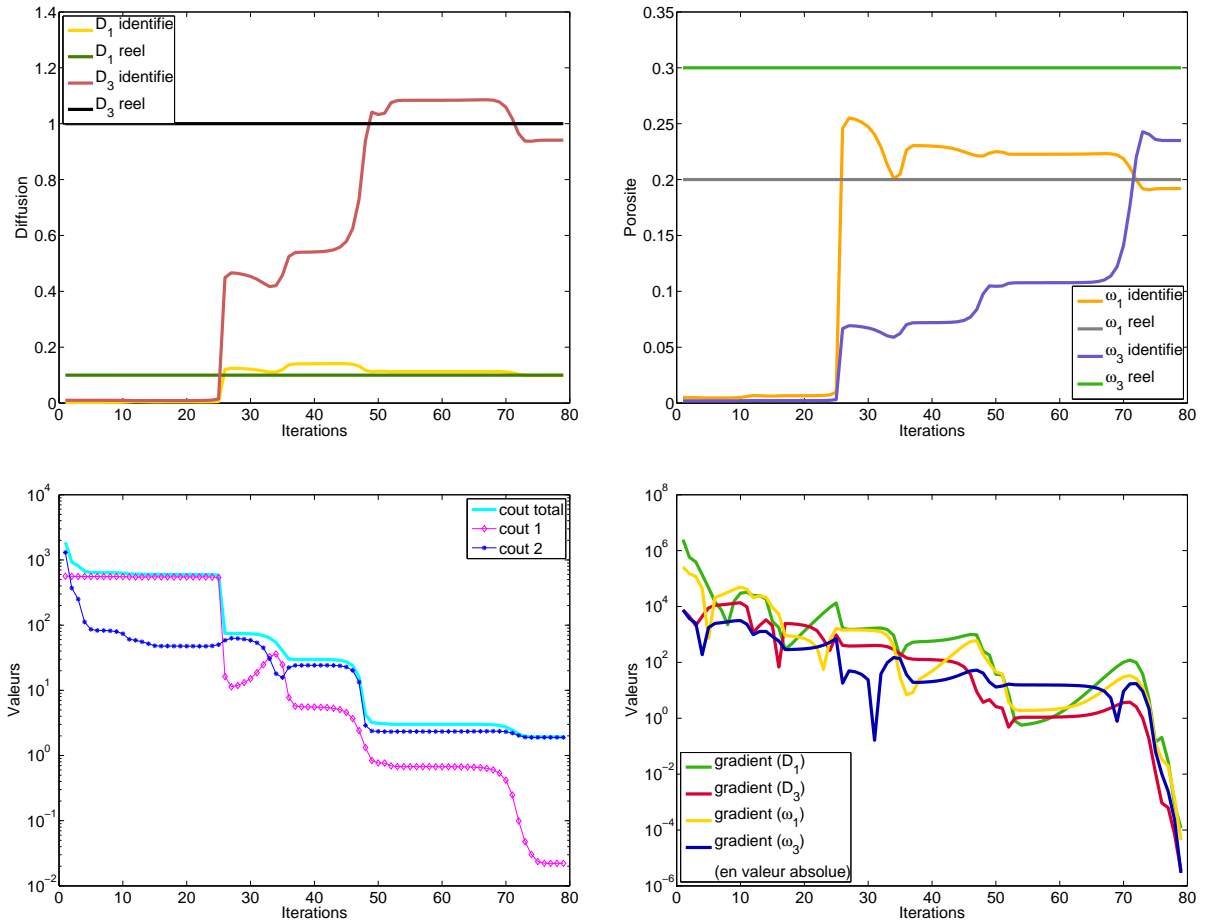


FIG. 5.4 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

Dans le cas d'un bruit de 5%, l'algorithme a convergé en 80 itérations et les résultats (tableau 5.9) montrent que les paramètres estimés sont proches des paramètres synthétiques (excepté pour ω_3). L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.5.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,0983638494	0,1
D'_3	1.10^{-2}	0,9025316218	1,0
ω_1	5.10^{-3}	0,1865508366	0,2
ω_3	2.10^{-3}	0,1929961117	0,3

TAB. 5.9 – Identification dans les zones 1 & 3 avec un bruit de 5%

La zone endommagée Ω_3 étant très mal observée, on constate qu'en augmentant le bruit les paramètres associés à cette zone sont moins bien identifiés que ceux de la zone 1.

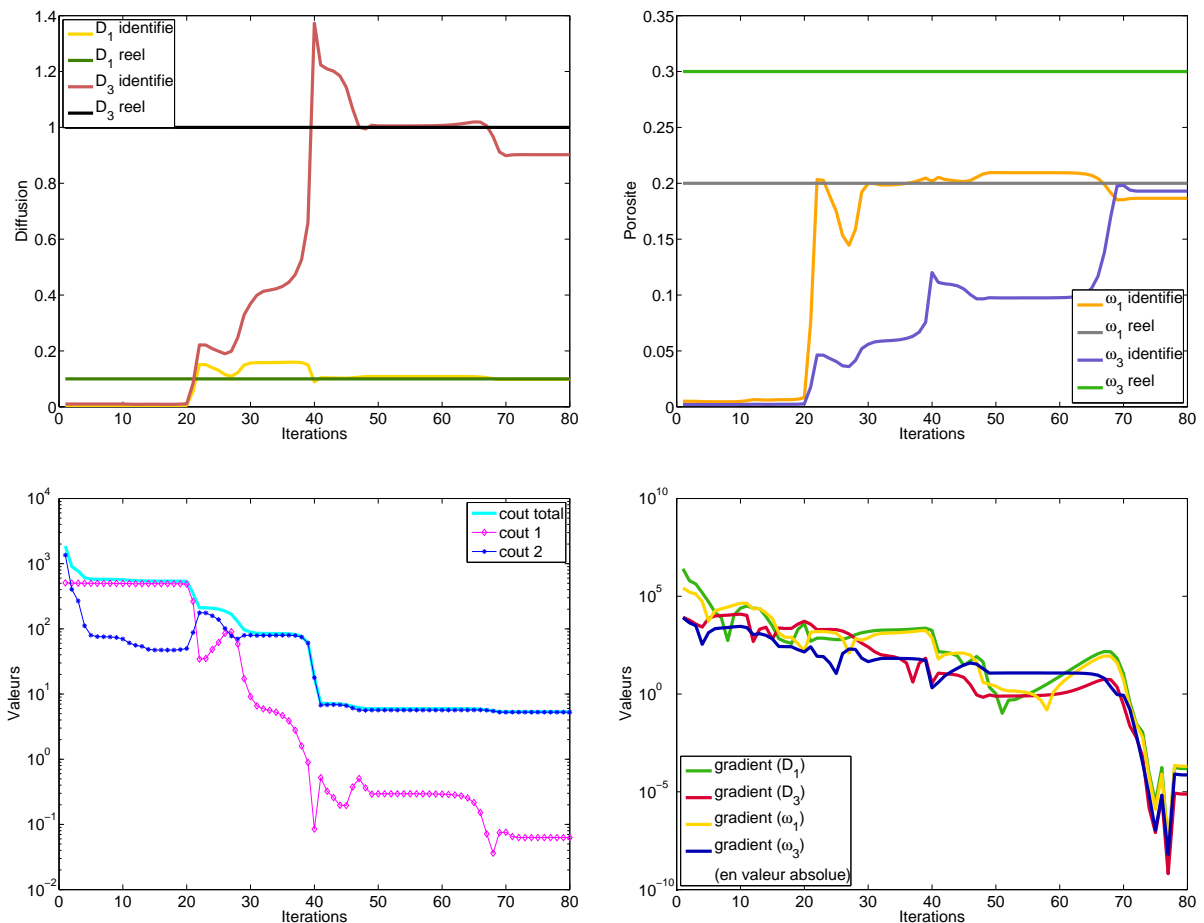


FIG. 5.5 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

Bruit relatif

Dans le paragraphe précédent, nous avons fait nos tests avec un bruit généré “globalement” ce qui pouvait donner des mesures non significatives dans la zone loin du colis (concentration faible, bruit grand). Dans cette section, nous générons un bruit par rapport à la concentration locale : si par exemple le bruit est de 5 %, on aura pour une concentration de 1 mol/m^3 un bruit de l'ordre de 0.05 sur la concentration et pour une concentration de 10^{-4} mol/m^3 un bruit de l'ordre de 0.05×10^{-4} . De cette manière, lorsqu'on a peu de concentration on génère peu de bruit et ainsi l'influence du bruit est à peu près la même partout ce qui devrait dégrader les performances dans toutes les zones de la même façon. On présente dans cette section plusieurs tests réalisés avec un bruit croissant.

Dans le cas d'un bruit de 3%, l'algorithme a convergé en 81 itérations et les résultats (tableau 5.10) montrent que la solution est bien approchée. On remarque que les coefficients dans le tableau ci-dessous sont mieux identifiés que dans le cas du bruit absolu sauf pour ω_3 . L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.6.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,1011234413	0,1
D'_3	1.10^{-2}	0,9721319375	1,0
ω_1	5.10^{-3}	0,2028821108	0,2
ω_3	2.10^{-3}	0,2244285879	0,3

TAB. 5.10 – Identification dans les zones 1 & 3 avec un bruit de 3%

Remarque 5.6 *Toutes les simulations réalisées montrent que dès qu'il y a du bruit (absolu ou relatif), ω_3 est difficilement identifiable.*

Dans le cas d'un bruit de 5%, l'algorithme a convergé en 85 itérations et les résultats (tableau 5.11) montrent que la solution est bien approchée. On peut observer que les valeurs des paramètres identifiés sont encore très satisfaisantes (toujours excepté pour ω_3). L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.7.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,1018767983	0,1
D'_3	1.10^{-2}	0,9552432582	1,0
ω_1	5.10^{-3}	0,2049359853	0,2
ω_3	2.10^{-3}	0,1729395267	0,3

TAB. 5.11 – Identification dans les zones 1 & 3 avec un bruit de 5%

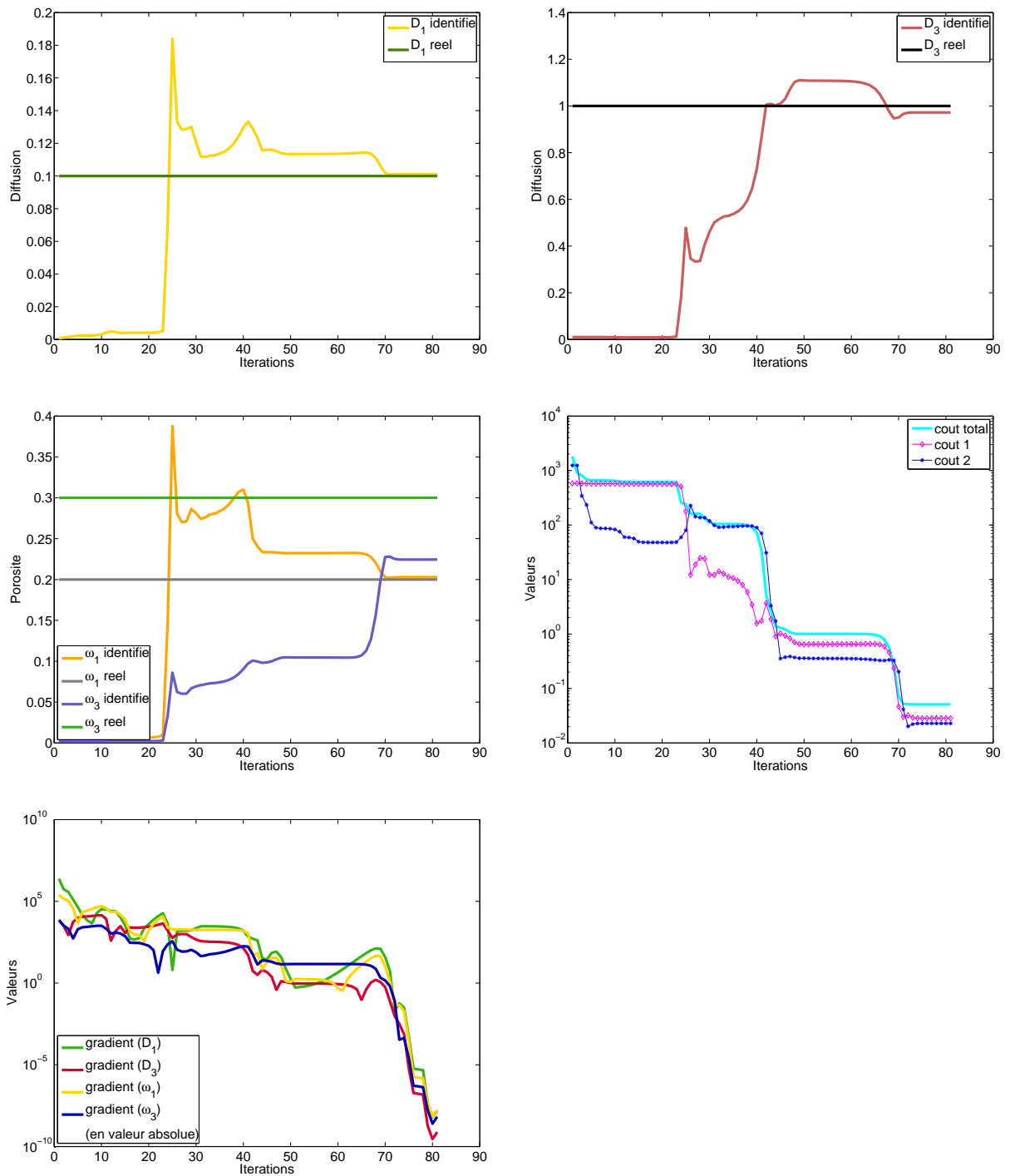


FIG. 5.6 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

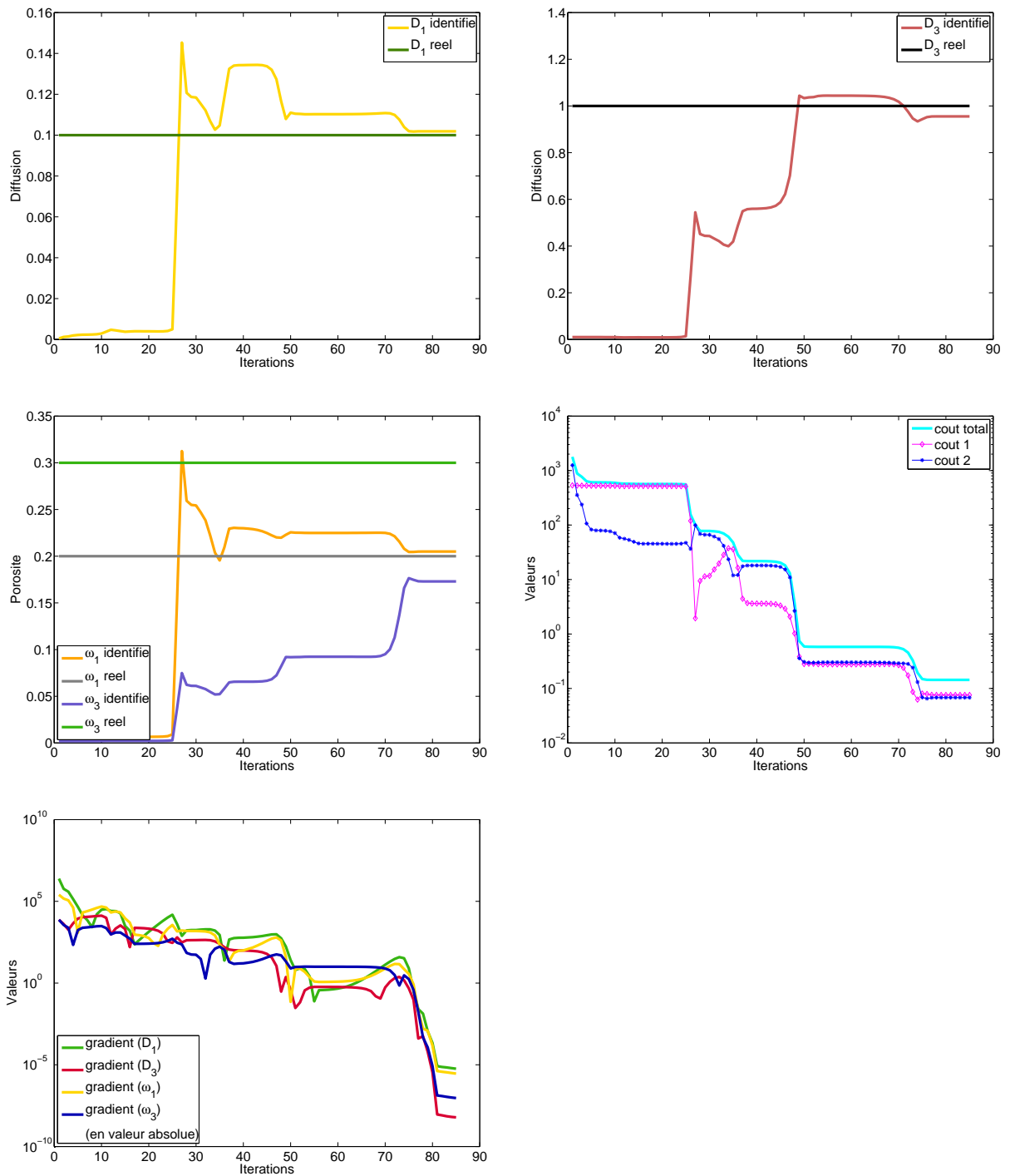


FIG. 5.7 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

Dans les deux tests qui suivent, nous avons voulu voir les limites de l'algorithme en augmentant le bruit de façon significative : le premier test a été réalisé avec un bruit de 10% et le second avec un bruit de 20%.

Dans le cas d'un bruit de 10%, l'algorithme a tourné pendant 84 itérations et les résultats (tableau 5.12) montrent que la solution reste assez bien approchée. On peut observer que les valeurs des paramètres identifiés sont encore satisfaisantes (toujours excepté pour ω_3).

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,1037084265	0,1
D'_3	1.10^{-2}	0,9223621477	1,0
ω_1	5.10^{-3}	0,2103435819	0,2
ω_3	2.10^{-3}	0,0491742038	0,3

TAB. 5.12 – Identification dans les zones 1 & 3 avec un bruit de 10%

L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.8.

Dans le cas d'un bruit de 20%, l'algorithme a itéré 65 fois sans toutefois converger vers les bonnes valeurs. On observe que les paramètres dans le tableau 5.13 sont beaucoup moins bien identifiés par rapport aux autres simulations, ce qui n'est pas très étonnant avec un bruit aussi élevé. En ce qui concerne ω_3 , sa valeur a changé au cours des itérations pour se bloquer à la borne inférieure donné au minimiseur.

Paramètres	initiaux	identifiés	réels
D'_1	5.10^{-4}	0,0867632329	0,1
D'_3	1.10^{-2}	0,6758693924	1,0
ω_1	5.10^{-3}	0,1737230736	0,2
ω_3	2.10^{-3}	1.10^{-4}	0,3

TAB. 5.13 – Identification dans les zones 1 & 3 avec un bruit de 20%

L'évolution des paramètres au cours des itérations, ainsi que celle de la fonction coût et du gradient, est présentée sur la figure 5.9.

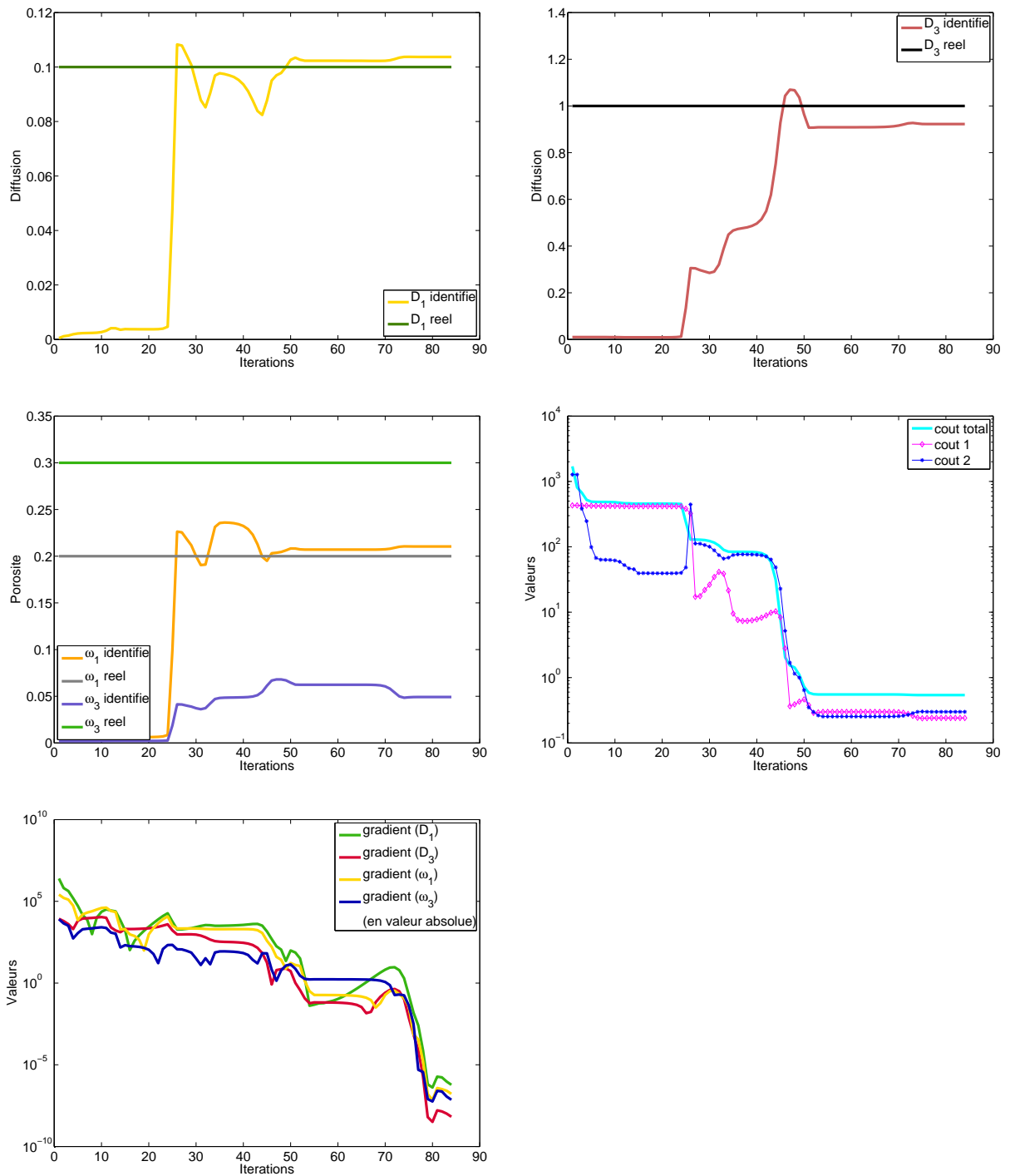


FIG. 5.8 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

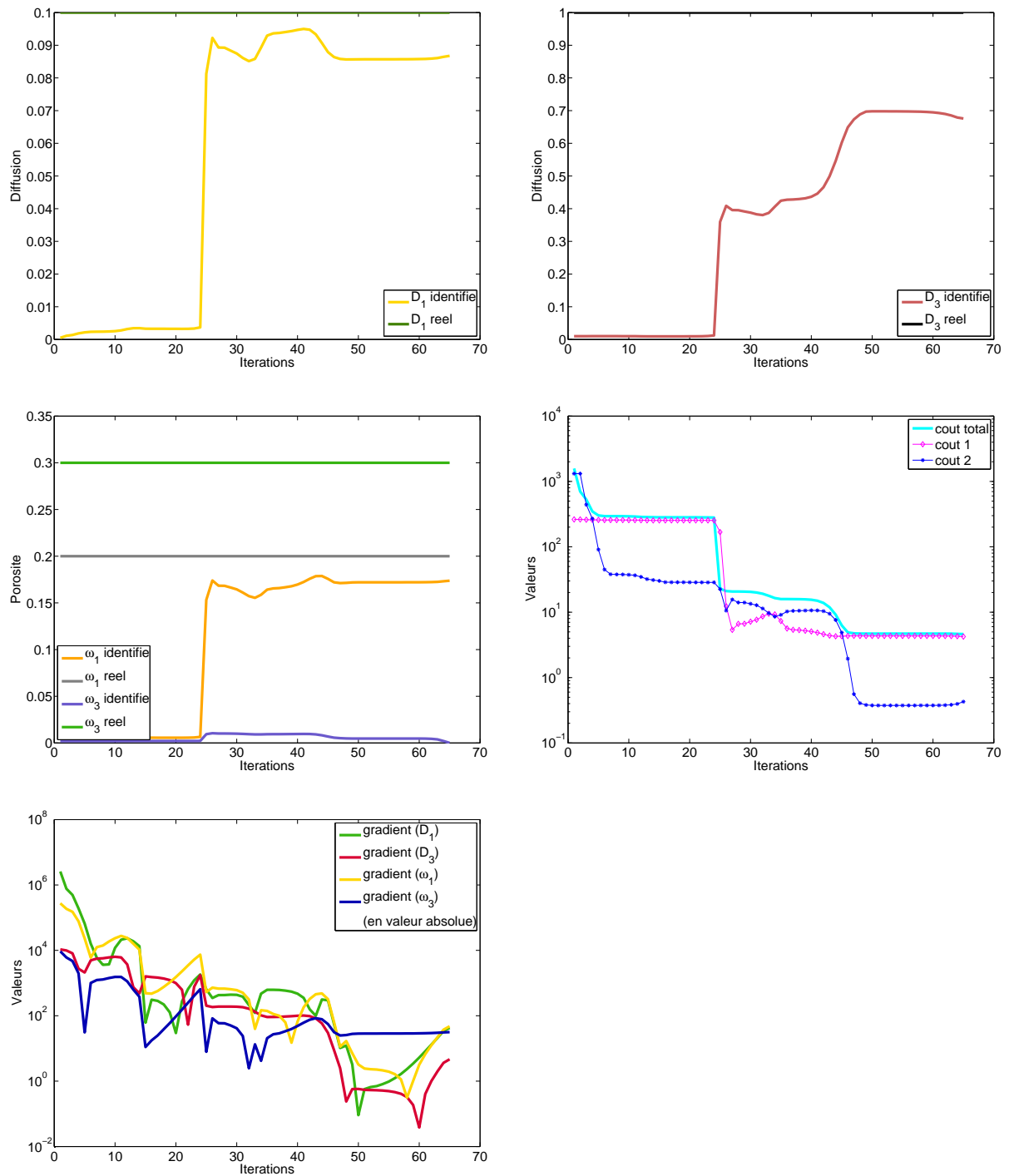


FIG. 5.9 – En haut à gauche : évolution des 2 paramètres de diffusion. En haut à droite : évolution des 2 paramètres de porosité. En bas à gauche : évolution de la fonction coût en échelle semi-log (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan). En bas à droite : évolution des 4 composantes du gradient en échelle semi-log.

5.3 Conclusion des tests

Nous avons choisi d’approcher les paramètres recherchés par des fonctions constantes par zones, ce qui semble être dans notre cas assez judicieux. Ces tests synthétiques ont permis de valider l’approche par état adjoint dont la partie théorique a été développée au chapitre 3. Ils ont également mis en évidence la nécessité d’adimensionner les paramètres pour les deux cas-tests, l’importance des observations à l’instant final pour le cas-test “colis” et la difficulté d’identifier la porosité dans la zone endommagée pour le cas-test “réel”.

Ces cas-tests ont été choisis en liaison avec les équipes de l’Andra afin de représenter des problèmes physiques réels. Les résultats dans le cas de données non bruitées se sont révélés très satisfaisants aussi bien pour le cas-test “colis” que pour le cas-test “réel”. En ce qui concerne l’identification avec données bruitées, les résultats obtenus pour le cas-test “colis” sont en adéquation avec les observations disponibles c’est-à-dire qu’on a une bonne identification des coefficients dans la zone la plus observée et une identification incorrecte dans la zone la moins observée. Pour le cas-test “réel”, les résultats obtenus avec un bruit absolu sont corrects et ils sont légèrement meilleurs avec un bruit relatif. Les tests ont prouvé la stabilité de la solution du problème inverse.

Pour terminer, nous abordons dans ce paragraphe l’analyse de sensibilité. Nous parlerons tout d’abord de l’analyse de sensibilité des mesures par rapport aux paramètres. A cet effet, on peut citer Clément et al. [41] pour l’analyse de sensibilité d’une équation de diffusion 1D, l’article de Marchand et al. [74] ainsi que la thèse de Marchand [73] pour une analyse de sensibilité probabiliste (type Monte Carlo) et déterministe pour un modèle d’écoulement 3D et un modèle écoulement-transport 2D. En ce qui nous concerne, nous avons pu quantifier la sensibilité des paramètres dans nos différents cas-tests. En effet, si on regarde les évolutions du gradient de la fonction coût par rapport aux paramètres (figures 5.1 à 5.9), on remarque que le problème est plus sensible (et donc dépendant) à la diffusion qu’à la porosité. On voit également que les deux paramètres sont plus sensibles dans la roche Ω_1 que dans la zone endommagée Ω_3 , on a d’ailleurs eu de meilleurs résultats d’identification dans la zone 1 que dans la zone 3.

Chapitre 6

Identification de zones

Sommaire

6.1	Introduction	92
6.2	Etude numérique	93
6.2.1	Premiers tests	93
6.2.1.1	Première expérience	94
6.2.1.2	Deuxième expérience	97
6.2.2	Cas concentrique	100
6.2.2.1	Données non bruitées	100
6.2.2.2	Données bruitées	105
6.3	Conclusion des tests	114

Dans le chapitre 3, nous nous sommes intéressés à l'estimation de paramètres connaissant la géométrie exacte du domaine. On cherche ici, partant d'une configuration géologique modifiée, à retrouver la configuration géologique exacte (celle qui a permis de générer les données synthétiques du problème direct) en identifiant les paramètres qui ont pour support ces zones. En regroupant ensuite par isovalues les coefficients identifiés, nous aurons la configuration recherchée. Ces tests sont très importants puisqu'en pratique le nombre et la position exacte des sous-domaines sont inconnus.

En ce qui concerne les données (synthétiques), nous avons utilisé pour les observations les valeurs de concentration générées avec la zonation exacte du domaine dans les cas sans bruit et avec bruit. Pour le bruit, nous avons également utilisé le bruit généré avec la zonation exacte du domaine.

Dans ce chapitre, nous présentons tout d'abord les diverses méthodes de résolution des problèmes d'identification de zones. Nous exposons ensuite les résultats numériques obtenus lors de nos différents tests. La dernière section fait une synthèse du travail accompli.

6.1 Introduction

Concernant l'identification de zones, on peut citer plusieurs méthodes, qui ont d'ailleurs déjà fait l'objet de recherches qui se poursuivent au sein du thème «Problèmes inverses et analyse de sensibilité» du GNR MoMaS :

- méthode des indicateurs de raffinement qui consistent à découper de façon «optimale» une des zones en deux parties distinctes,
- méthode d'optimisation de forme basée sur la méthode des lignes de niveau ou level sets,
- méthode d'optimisation topologique, qui consiste à étudier l'évolution d'un critère en ajoutant un trou au domaine.

La méthode des indicateurs de raffinement et de déraffinement (cf. les travaux de Ben Ameer et al. [15, 16, 17], Chavent & Bissel [36], Hayek et al. [60]) consiste à introduire une partition du domaine appelée partition de paramétrisation dont les parties de formes diverses correspondent à des zones où les coefficients à estimer sont constants. Les degrés de liberté des paramètres à identifier sont en nombre égal au nombre de parties de cette partition. A chaque paramètre ajouté à la paramétrisation initiale correspond un indicateur de raffinement. Ces indicateurs, qui s'obtiennent par des calculs simples (utilisant le gradient de la fonction coût), sont en fait des multiplicateurs de Lagrange associés à des contraintes relatives au problème considéré. A ces indicateurs de raffinement correspondent également des indicateurs de déraffinement qui permettent de refusionner des zones au cours de l'optimisation si le découpage a été trop fin.

La méthode d'optimisation de forme, comme utilisée par Allaire et al. dans [4, 5, 6], consiste à rechercher une fonction caractéristique au lieu de rechercher des zones, les deux problèmes étant équivalents. Cependant, comme ce problème est non différentiable, une solution consiste à remplacer la fonction caractéristique qui prend ses valeurs dans $\{0, 1\}$ par une fonction qui prend ses valeurs dans l'intervalle $[0, 1]$, la faire évoluer au fil de l'optimisation, puis à seuiller ou prendre une ligne de niveau (level set, par exemple $1/2$ si on travaille dans $[0, 1]$) pour décider si une maille est dans une zone ou l'autre. On peut également citer comme référence Bendsøe [18], Wang et al. [98].

Concernant la méthode d'optimisation topologique [31], c'est un peu la même idée que pour l'optimisation de forme type level set, sauf qu'on garde les fonctions caractéristiques dans $\{0, 1\}$ mais on rend le problème différentiable en ne considérant que des perturbations

infinitésimales (par exemple une boule de rayon ε). On ne fait varier les domaines, ou la fonction caractéristique, que d'une maille à la fois et on calcule la variation du critère (la fonction coût) par rapport à cette petite variation et on récupère un gradient, dit topologique. Là où il est le plus négatif, on fera diminuer de façon optimale le critère en faisant changer une maille de zone (cf. Garreau et al. [51] et Guillaume & Sid Idris [58]). Ce genre de technique marche très bien pour la détection de zones homogènes dans les images (Auroux & Masmoudi [13]), pour la restauration d'images (Belaid et al. [14]) ou pour la détection d'inhomogénéités dans un matériau (Amstutz et al. [7]).

Comme nous l'avons expliqué dans le chapitre 1, nous avons privilégié dans ce travail, la mise au point d'un outil adaptable à différentes situations. C'est pour cette raison que nous avons utilisé la même méthode que pour l'identification de paramètres et que nous n'avons pas retenu la méthode des indicateurs de raffinement et les méthodes d'optimisation de formes car il fallait pouvoir remailler le domaine (ou faire évoluer les zones) à chaque itération ce qui n'est pas prévu dans le code Traces où la génération du maillage est faite "à la main" et la gestion des zones est déclarée dès le départ en dur dans le code. Nous avons donc travaillé sur un maillage fixe et un nombre de zones fixés. Ceci étant, les trois méthodes présentées dans ce paragraphe sont tout à fait adaptées à notre problème et pourraient faire l'objet d'un futur travail (cf. chapitre 7).

6.2 Etude numérique

Dans ce chapitre, nous travaillons avec la géométrie du cas-test "réel". Nous avons dû modifier le fichier de maillage en fonction de la zonation de nos différents tests. Le cadre de travail est le même que pour l'identification de paramètres.

6.2.1 Premiers tests

Une première action consiste, à partir du maillage actuel, à modifier la zone endommagée Ω_3 (en l'agrandissant) et à considérer que tous les éléments de cette nouvelle zone ont des paramètres de diffusion et de porosité différents puis de lancer l'identification sur tous ces paramètres afin de voir comment le système se recalcule (ce qui revient à identifier les coefficients par maille).

La nouvelle zone endommagée comprend 440 mailles (et donc 440 zones), il y a ainsi 880 paramètres à estimer (comme pour l'identification de paramètres on identifie pas les coefficients dans le colis). La configuration géologique exacte du domaine est donnée par la figure 6.1 et la nouvelle zonation par la figure 6.2. Sur la figure 6.1, on voit trois zones : la zone 1 (ou Ω_1) dite zone "loin", la zone 2 (ou Ω_2) correspondant au colis et la zone 3 (ou Ω_3) représentant la zone endommagée. Sur la figure 6.2, on a toujours la zone 1 (mais plus petite que sur la figure 6.1), le colis en zone 2 et la nouvelle zone endommagée (élargie sur une partie de l'ancienne zone 1) qui comprend 440 zones.

Dans les deux tests qui suivent, on va donc partir de la configuration modifiée (figure 6.2) en identifiant les coefficients par maille pour essayer de retrouver la zonation exacte (figure 6.1).

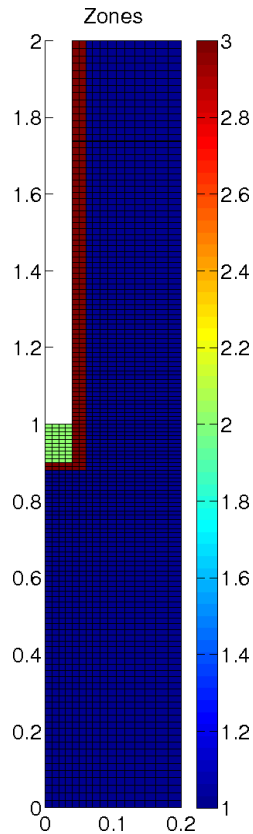


FIG. 6.1 – Configuration exacte

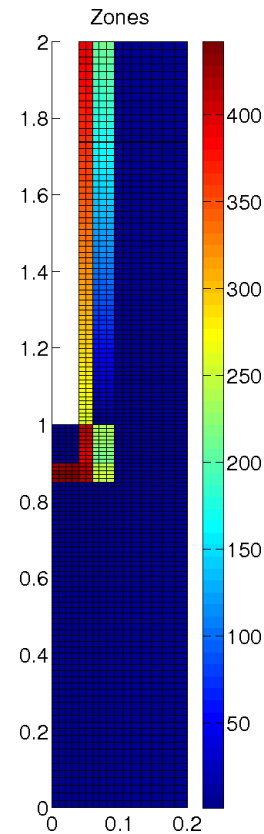


FIG. 6.2 – Configuration modifiée

6.2.1.1 Première expérience

Nous avons initialisé les coefficients avec les valeurs ci-dessous.

Diffusion	initiale	Porosité	initiale
D'_1	0, 2	ω_1	0, 15
D'_3 à D'_{442}	0, 6	ω_3 à ω_{442}	0, 25

TAB. 6.1 – Valeurs initiales des paramètres

Etant donné qu'il y a beaucoup de paramètres à identifier, la convergence de l'algorithme est assez lente (même en partant de valeurs initiales proches des valeurs exactes). On voit sur les figures 6.3 et 6.4 que l'identification des paramètres (et donc des zones) est en bonne voie. Cependant, les évolutions de la fonction coût et du gradient sur les figures 6.5 et 6.6 montrent que l'algorithme n'a pas terminé la minimisation et nécessiterait des itérations supplémentaires. Nous avons effectivement bloqué la minimisation à 1000 simulations (pour des raisons de coût et de temps de calcul) et N2QN1 s'est arrêté lorsqu'il a atteint cette borne.

On remarque également qu'au-dessus de la droite horizontale γ_1 (cf. chapitre 2 figure 2.6), on a aucune information sur la concentration. En effet, on a simplement des observations au temps final pour les zones 1 & 3 (zone endommagée et roche), il est donc possible d'estimer les coefficients qui sont dans un voisinage proche de la droite mais il sera quasiment impossible d'identifier ceux qui se trouvent au-dessus (situés à peu près entre 1, 2 et

2 m de hauteur). On observe effectivement sur les figures 6.3 et 6.4 qu’au voisinage du colis on retrouve des valeurs de diffusion et de porosité proches de la réalité, ce qui montre que l’algorithme fonctionne bien dans les parties observées. Pour étendre aux zones non observées, il faut ajouter de l’information a priori pour régulariser le problème : c’est ce qu’on fait dans le paragraphe 6.2.2 en précisant que les zones recherchées sont “cylindriques”, ce qui permet d’étendre l’identification aux zones plus loin du colis.

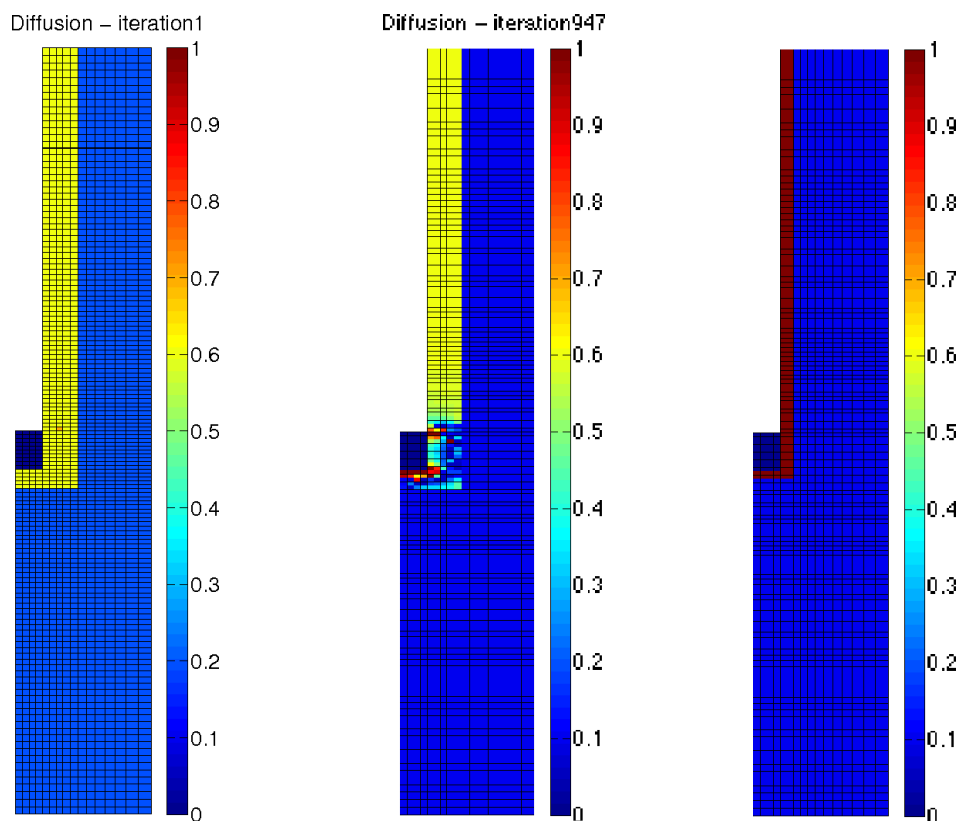


FIG. 6.3 – Diffusion à l’itération 1, Diffusion à l’itération 947, Diffusion réelle

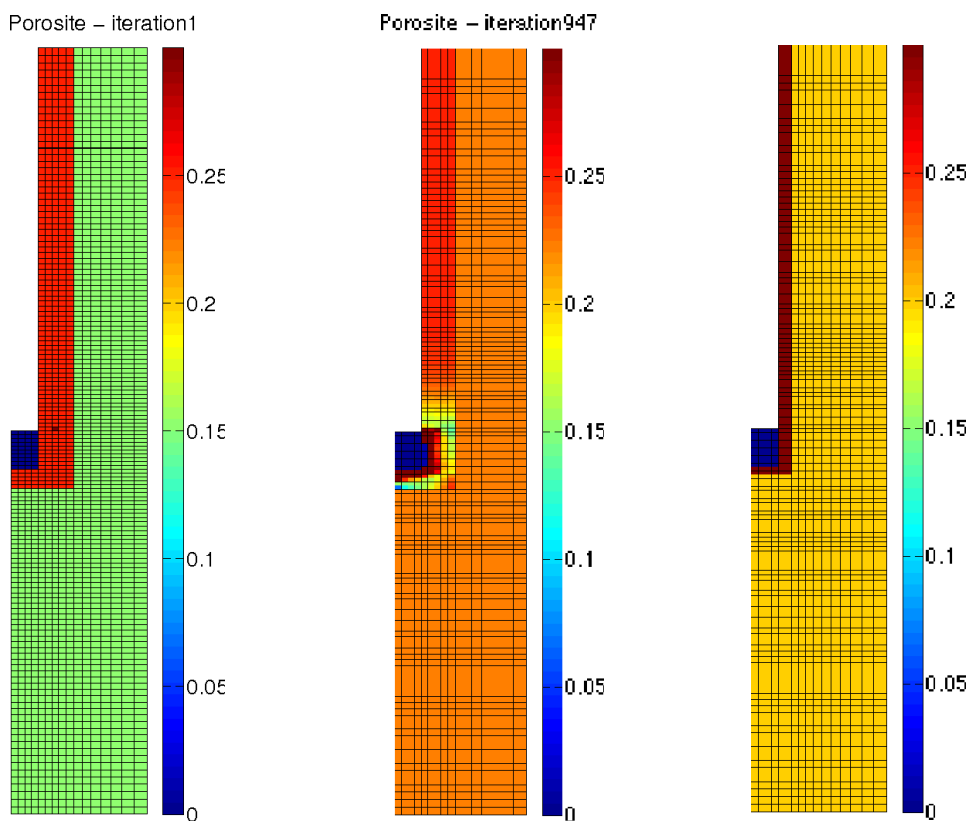


FIG. 6.4 – Porosité à l'itération 1, Porosité à l'itération 947, Porosité réelle

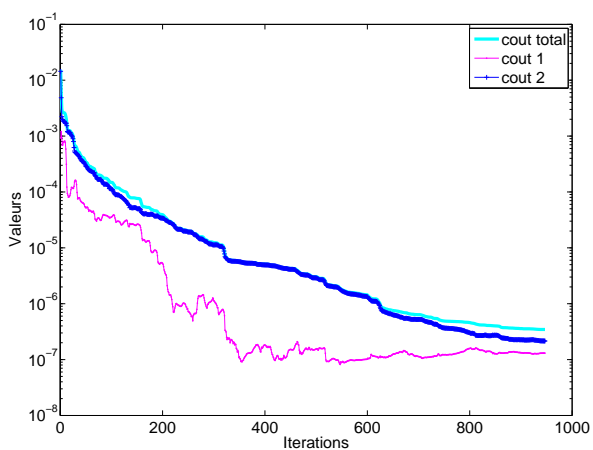


FIG. 6.5 – Evolution de la fonction coût (la partie temporelle est en magenta, la partie correspondant à l'état final est en bleu et le coût total est en cyan)

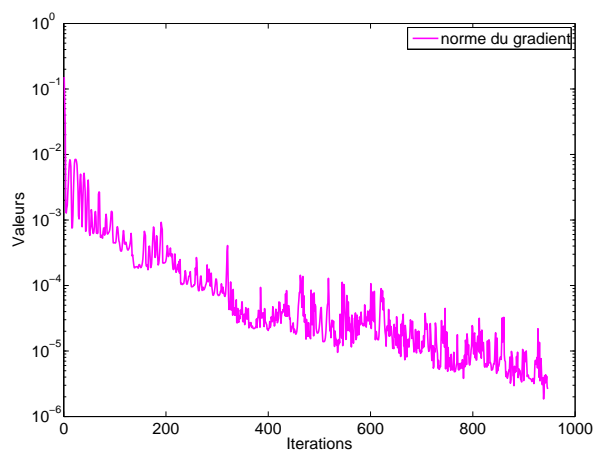


FIG. 6.6 – Evolution du gradient

6.2.1.2 Deuxième expérience

Nous sommes dans la même configuration que l'expérience précédente et nous avons initialisé les coefficients avec les valeurs ci-dessous.

Diffusion	initiale	Porosité	initiale
D'_1	0,2	ω_1	0,1
D'_3 à D'_{442}	0,7	ω_3 à ω_{442}	0,4

TAB. 6.2 – Valeurs initiales des paramètres

Nous avons laissé tourner l'algorithme de minimisation plus longtemps, le critère d'arrêt ayant été fixé à 3000 simulations. N2QN1 a itéré 2501 fois et s'est arrêté sur le nombre maximal de simulations permis. Comme pour la première expérience, il faudrait encore laisser évoluer l'algorithme car on voit sur les figures 6.9 et 6.10 que la minimisation n'est pas terminée. En effet, on observe qu'autour de la zone colis les valeurs des paramètres ne sont pas totalement identifiées mais qu'elles se rapprochent néanmoins des valeurs réelles.

En comparant les évolutions de la fonction coût et du gradient (figures 6.7 et 6.8) à celles du premier test (figures 6.5 et 6.6), on constate que la minimisation s'est poursuivie correctement de l'itération 947 à 2501.

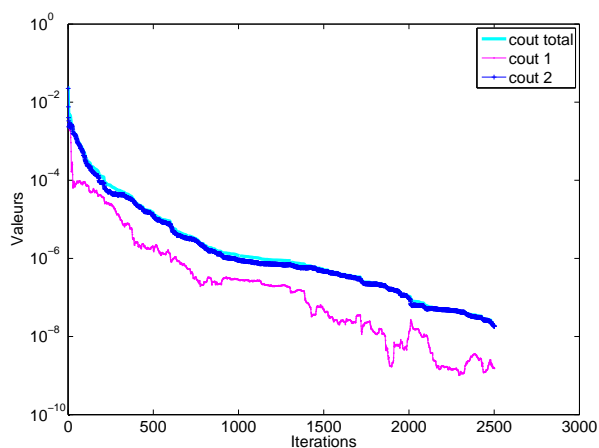


FIG. 6.7 – Evolution de la fonction coût

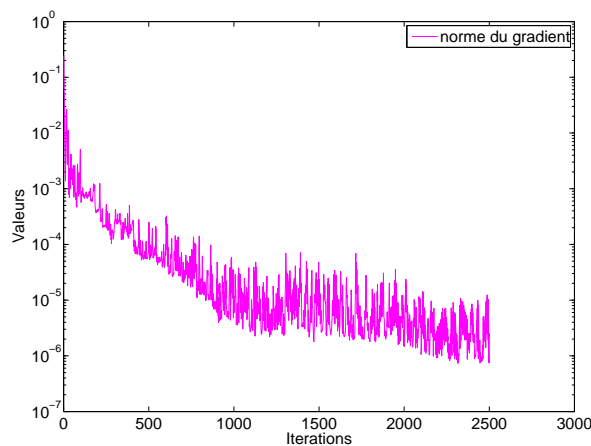


FIG. 6.8 – Evolution du gradient

Ce test a permis de montrer que l'identification des paramètres par maille était possible. Le processus est cependant très lent à cause du grand nombre de coefficients à identifier et du caractère mal posé du problème. Il faudrait sûrement régulariser la fonction coût pour avoir un problème bien posé et accélérer la convergence. La régularisation ne pourra venir que des informations physiques dont on dispose sur le problème (cf. le paragraphe suivant 6.2.2).

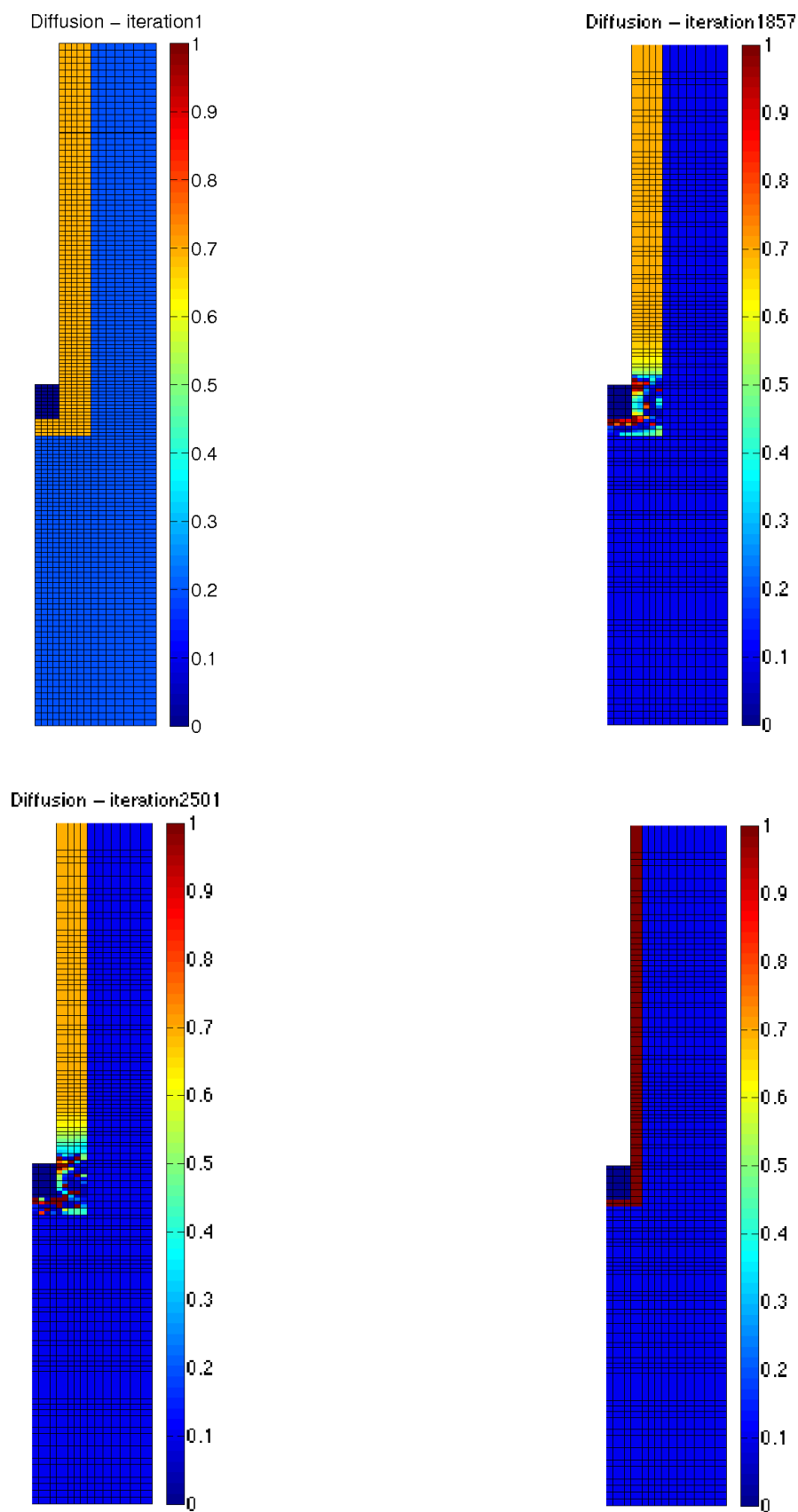


FIG. 6.9 – En haut à gauche : Diffusion à l'itération 1. En bas à droite : Diffusion à l'itération 1857. En bas à gauche : Diffusion à l'itération 2501. En bas à droite : Diffusion réelle.

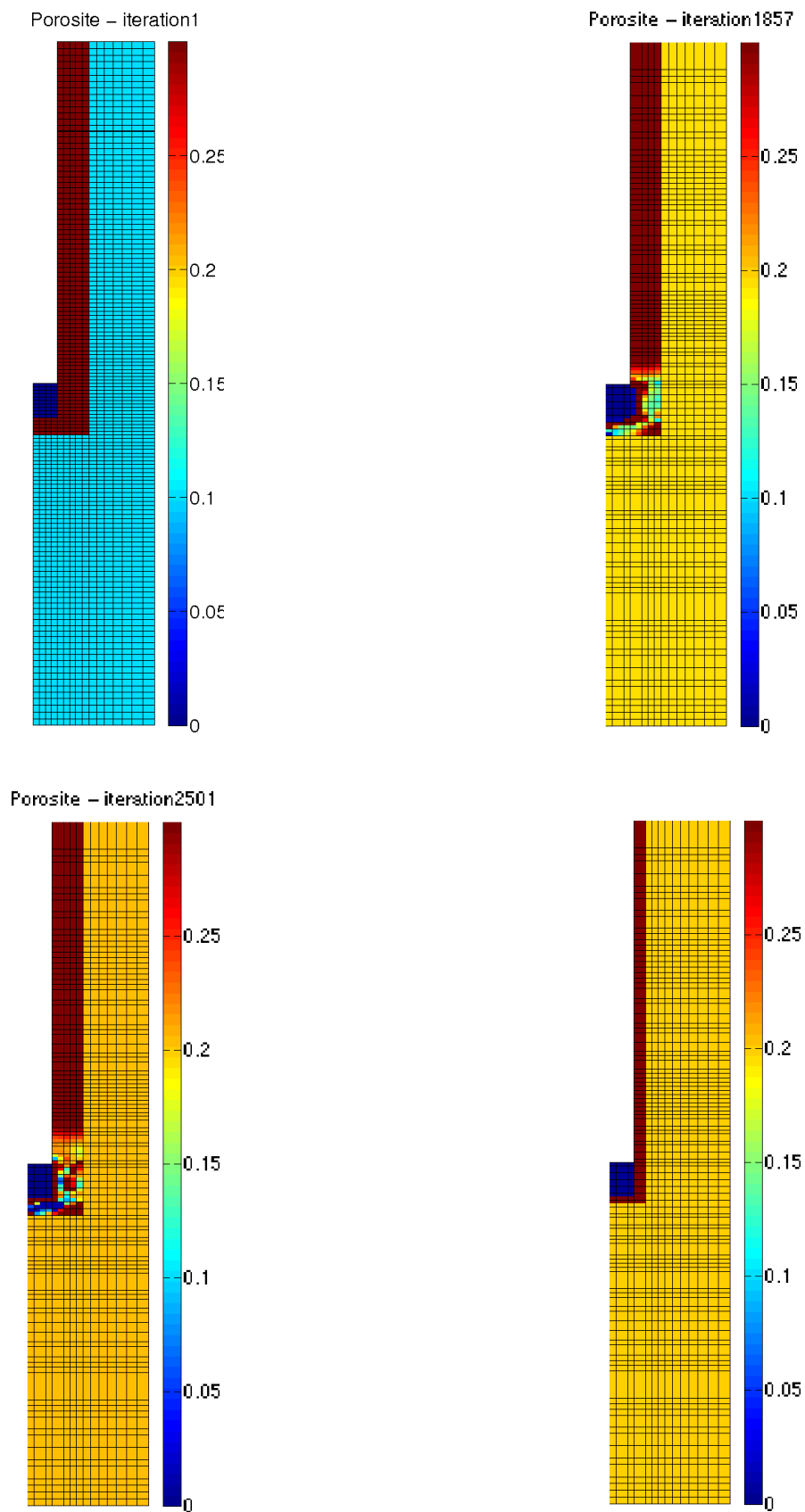


FIG. 6.10 – En haut à gauche : Porosité à l'itération 1. En bas à droite : Porosité à l'itération 1857. En bas à gauche : Porosité à l'itération 2501. En bas à droite : Porosité réelle.

6.2.2 Cas concentrique

Afin de prendre en compte la réalité du milieu et le type de modélisation qui repose sur la définition de “couches” à peu près concentriques (structure radiale) autour du creusement, nous avons mis au point avec les équipes de l’Andra un cas-test avec 7 zones dont 5 sont concentriques (figure 6.12). On peut voir sur la figure 6.12 que la zone loin est toujours rétrécie par rapport à la figure 6.11 et porte le numéro 1, le colis est toujours en zone 2 et l’ancienne zone endommagée élargie correspond aux zones concentriques 3 à 7.

D’un point de vue physique, il y a une sorte de symétrie de révolution autour du forage, c’est pourquoi on est amené à chercher des zones de cette forme. De plus, il y a peu de paramètres donc on peut essayer d’identifier les coefficients de cette façon pour regrouper ensuite les zones par valeurs proches.

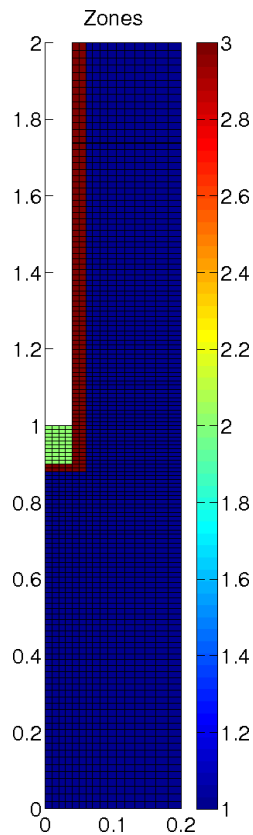


FIG. 6.11 – Configuration exacte

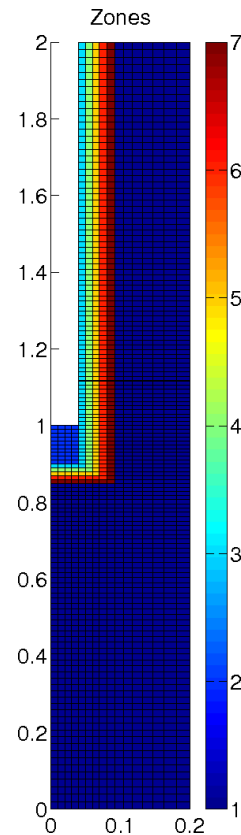


FIG. 6.12 – Configuration modifiée

6.2.2.1 Données non bruitées

Validation de la méthode

Afin de valider la méthode, nous commençons par un premier test avec des valeurs initiales proches des valeurs réelles. Il y a 12 paramètres à identifier : un paramètre de diffusion et de porosité par zone (sachant que comme dans le chapitre 5, la zone 2 est **fixée**). La recherche des paramètres de diffusion s’est réalisée entre les valeurs $10^{-5} m^2/an$ (borne inférieure) et $10^2 m^2/an$ (borne supérieure) et ceux de porosité entre 0,01 % et 1000 %.

Les résultats de l’inversion sont exposés dans le tableau 6.3 : pour chaque paramètre, la

deuxième colonne montre les valeurs initiales données au minimiseur, la troisième colonne présente les paramètres identifiés et la dernière colonne rappelle les valeurs réelles. Le tableau montre que l'identification a parfaitement fonctionné. L'algorithme a nécessité 179 itérations pour converger vers la bonne solution. Les évolutions de la fonction coût et du gradient au cours des itérations sont présentées sur les figures 6.15 et 6.16. La figure 6.13 (resp. 6.14) montre la diffusion (resp. la porosité) à l'itération 1 puis à l'itération finale et enfin la diffusion (resp. la porosité) réelle.

Diffusion	initiale	identifiés	réels	Porosité	initiale	identifiés	réels
D'_1	0,15	$9,9999999998 \cdot 10^{-2}$	0,1	ω_1	0,1	0,1919263062	0,2
D'_3	0,5	0,9999999999	1,0	ω_3	0,35	0,3000000000	0,3
D'_4	0,5	0,9999999999	1,0	ω_4	0,35	0,2999999999	0,3
D'_5	0,5	$9,9999999998 \cdot 10^{-2}$	0,1	ω_5	0,35	0,2000000000	0,2
D'_6	0,5	$9,9999999998 \cdot 10^{-2}$	0,1	ω_6	0,35	0,1999999999	0,2
D'_7	0,5	$9,9999999998 \cdot 10^{-2}$	0,1	ω_7	0,35	0,1999999999	0,2

TAB. 6.3 – Identification dans les zones 1, 3, 4, 5, 6 & 7

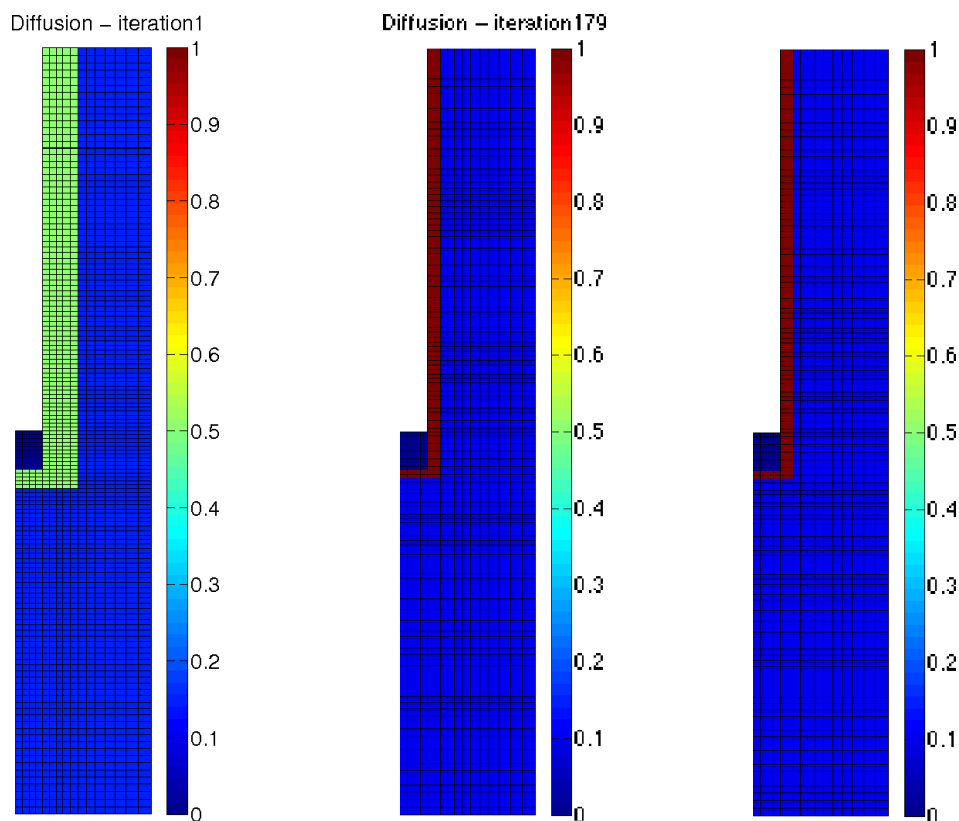


FIG. 6.13 – Diffusion à l'itération 1, Diffusion à l'itération 179, Diffusion réelle

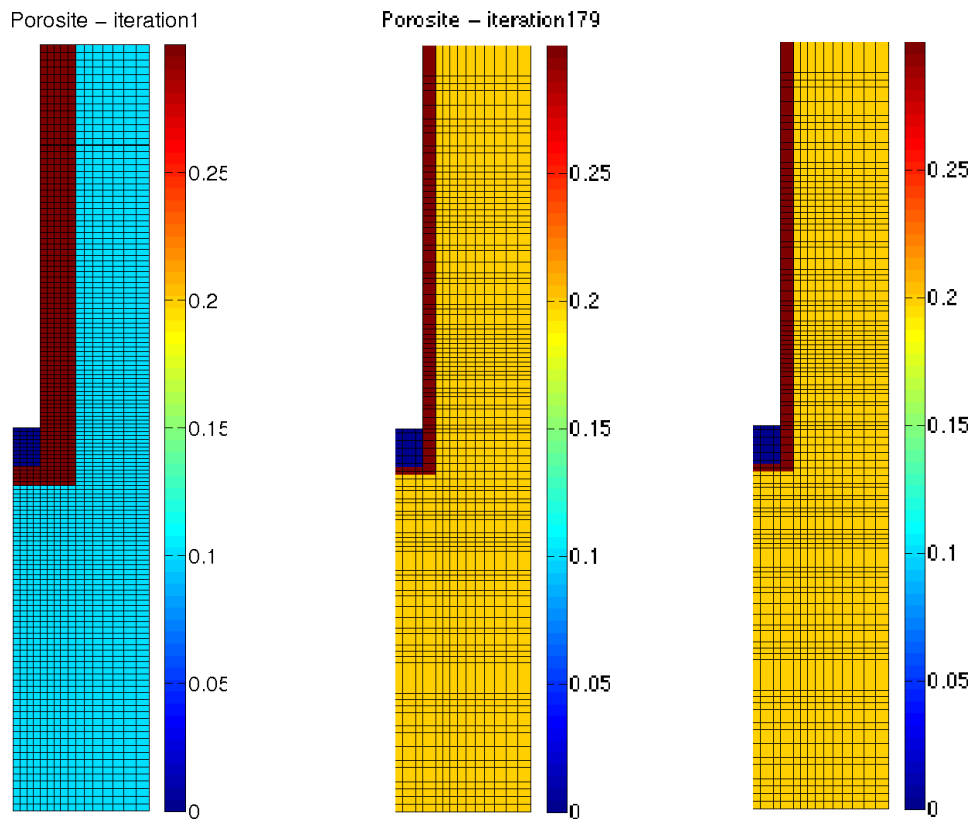


FIG. 6.14 – Porosité à l'itération 1, Porosité à l'itération 179, Porosité réelle

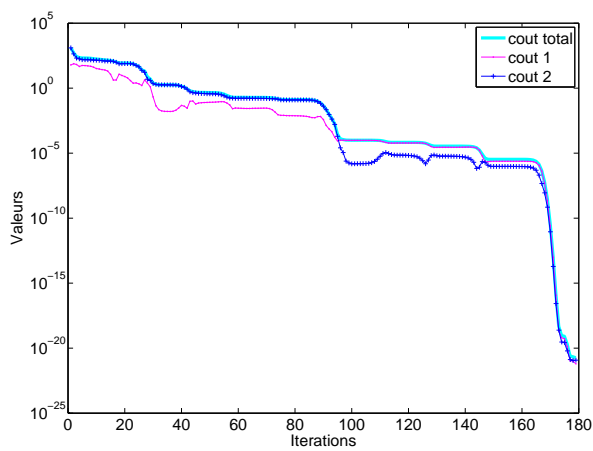


FIG. 6.15 – Evolution de la fonction coût

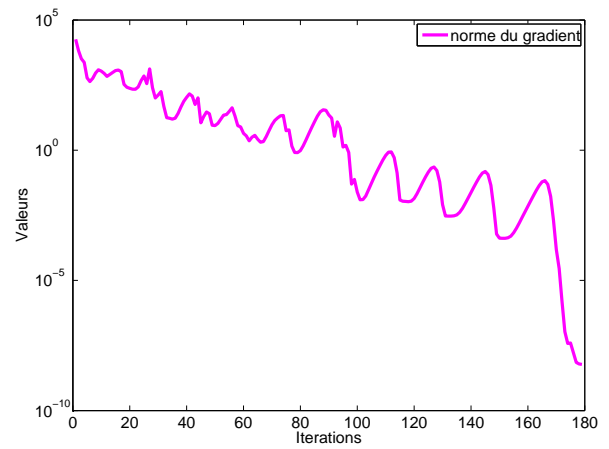


FIG. 6.16 – Evolution du gradient

Second test

Dans ce deuxième test, nous partons de données initiales assez éloignées des valeurs réelles. Le tableau 6.4 montre que l'identification a bien fonctionné (légèrement mieux pour la diffusion). L'algorithme a nécessité 270 itérations pour converger vers la bonne solution. Les évolutions de la fonction coût et du gradient au cours des itérations sont présentées sur les figures 6.19 et 6.20. La figure 6.17 (resp. 6.18) montre la diffusion (resp. la porosité) à l'itération 1 puis à l'itération finale et enfin la diffusion (resp. la porosité) réelle.

Diffusion	initiale	identifiés	réels	Porosité	initiale	identifiés	réels
D'_1	$5, 0.10^{-4}$	$9, 9992118456.10^{-2}$	0, 1	ω_1	$5, 0.10^{-3}$	0, 1999880262	0, 2
D'_3	$1, 0.10^{-2}$	0, 9999968747	1, 0	ω_3	$2, 0.10^{-3}$	0, 3079328776	0, 3
D'_4	$1, 0.10^{-2}$	1, 0013370402	1, 0	ω_4	$2, 0.10^{-3}$	0, 2846903435	0, 3
D'_5	$1, 0.10^{-2}$	$9, 9941973516.10^{-2}$	0, 1	ω_5	$2, 0.10^{-3}$	0, 2170407572	0, 2
D'_6	$1, 0.10^{-2}$	0, 1001078536	0, 1	ω_6	$2, 0.10^{-3}$	0, 1834338674	0, 2
D'_7	$1, 0.10^{-2}$	0, 1000217588	0, 1	ω_7	$2, 0.10^{-3}$	0, 2074470012	0, 2

TAB. 6.4 – Identification dans les zones 1, 3, 4, 5, 6 & 7

Les résultats sont légèrement moins bons (par rapport au premier test où les données initiales étaient proches des valeurs réelles) mais on distingue encore très nettement les deux “zones” recherchées parmi les cinq zones concentriques où la diffusion et la porosité sont quasi constantes.

D'autres tests ont été réalisés en partant de différentes données initiales. L'algorithme a également convergé vers la solution après environ 150 itérations pour des valeurs initiales proches des valeurs réelles et environ 250 itérations pour des valeurs initiales éloignées. On constate donc que l'algorithme est sensible à la position du point de départ (convergence plus rapide pour des valeurs initiales proches des valeurs réelles) et que l'algorithme a besoin de plus d'itérations pour converger par rapport à l'estimation de paramètres (où il n'y avait que 3 zones).

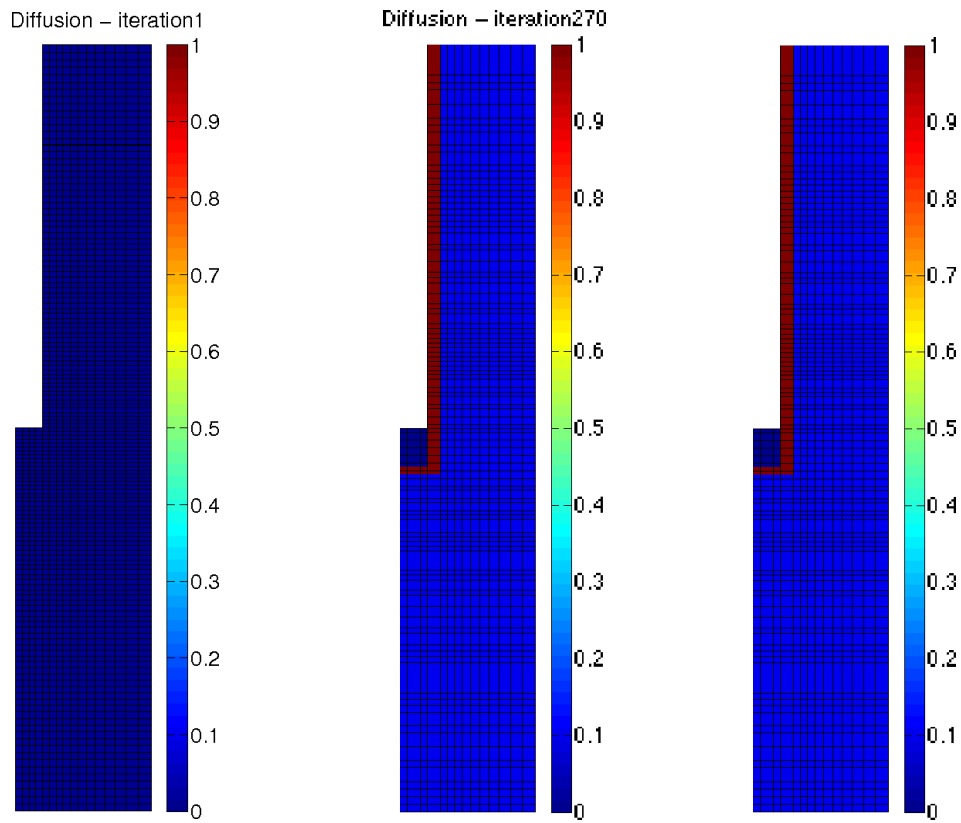


FIG. 6.17 – Diffusion à l'itération 1, Diffusion à l'itération 270, Diffusion réelle

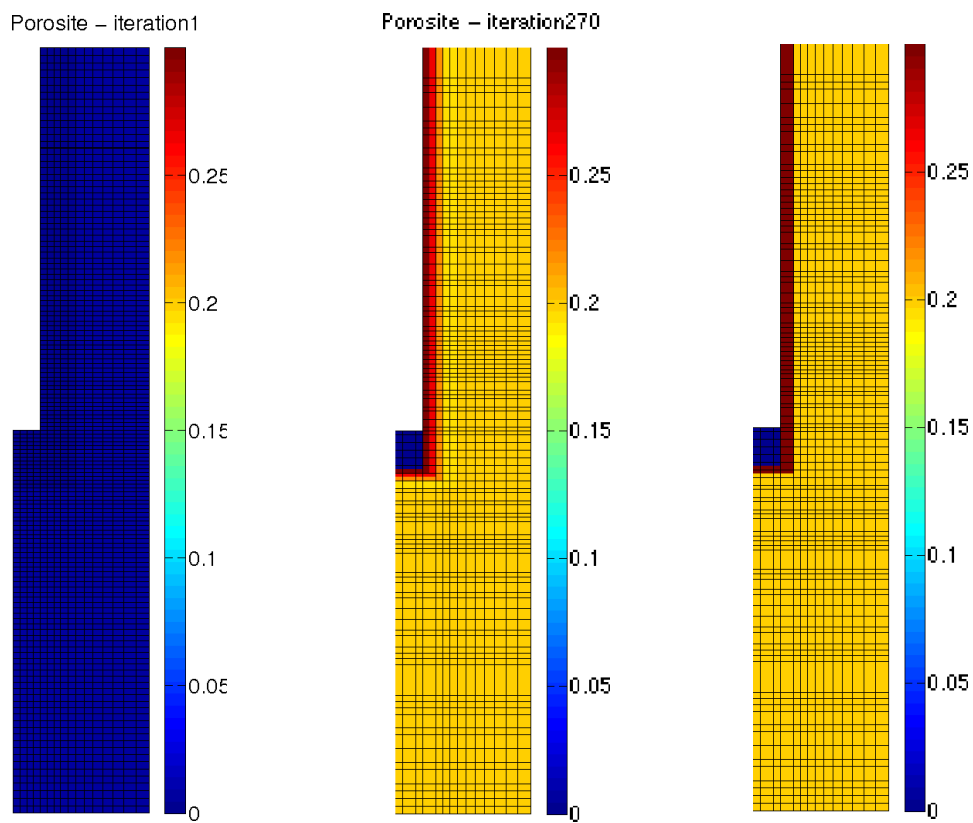


FIG. 6.18 – Porosité à l'itération 1, Porosité à l'itération 270, Porosité réelle

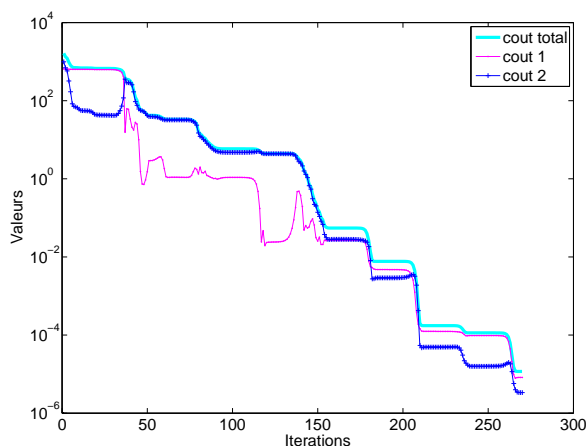


FIG. 6.19 – Evolution de la fonction coût

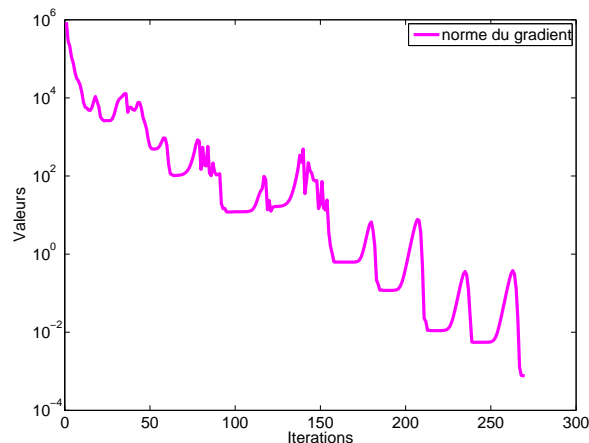


FIG. 6.20 – Evolution du gradient

6.2.2.2 Données bruitées

Dans ce paragraphe, nous utilisons le bruit “relatif” généré au chapitre 5 section 5.2.2.2 pour l’identification de paramètres. On présente plusieurs tests réalisés avec un bruit croissant.

Dans le cas d’un bruit de 3%, l’algorithme a itéré pendant 283 itérations et les résultats (tableau 6.5) montrent que la solution est bien approchée pour la diffusion. En effet, sur la figure 6.21 on voit bien à l’itération finale les trois zones recherchées. Pour la porosité (tableau 6.6), on constate que les coefficients ω_3 , ω_5 et ω_7 ne sont pas aussi bien identifiés que les autres ce qui ne permet pas d’arriver à la configuration géologique réelle (figure 6.22). Cependant, une réponse peut être apportée à cet inconvénient (cf. section 6.3).

Les évolutions de la fonction coût et du gradient sont présentées sur les figures 6.23 et 6.24.

Diffusion	initiale	identifiés	réels
D'_1	$5, 0.10^{-4}$	$9, 5819317524.10^{-2}$	0, 1
D'_3	$1, 0.10^{-2}$	$0, 9503963679$	1, 0
D'_4	$1, 0.10^{-2}$	$0, 9872350915$	1, 0
D'_5	$1, 0.10^{-2}$	$9, 8664397796.10^{-2}$	0, 1
D'_6	$1, 0.10^{-2}$	$9, 8621005507.10^{-2}$	0, 1
D'_7	$1, 0.10^{-2}$	$9, 7722807371.10^{-2}$	0, 1

TAB. 6.5 – Identification de la diffusion avec un bruit de 3%

Porosité	initiale	identifiés	réels
ω_1	$5, 0.10^{-3}$	0,1871781095	0,2
ω_3	$2, 0.10^{-3}$	$9,9999999999.10^{-5}$	0,3
ω_4	$2, 0.10^{-3}$	0,4455885768	0,3
ω_5	$2, 0.10^{-3}$	0,2190823028	0,2
ω_6	$2, 0.10^{-3}$	0,1482577474	0,2
ω_7	$2, 0.10^{-3}$	0,3208699000	0,2

TAB. 6.6 – Identification de la porosité avec un bruit de 3%

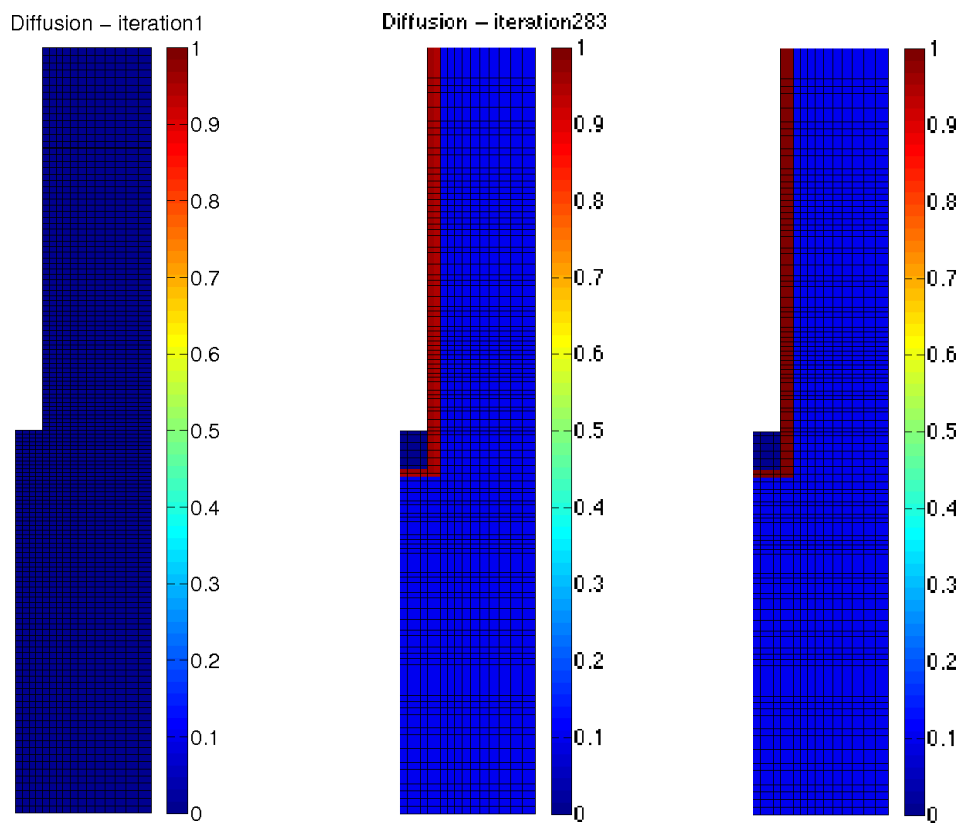


FIG. 6.21 – Diffusion à l'itération 1, Diffusion à l'itération 283, Diffusion réelle

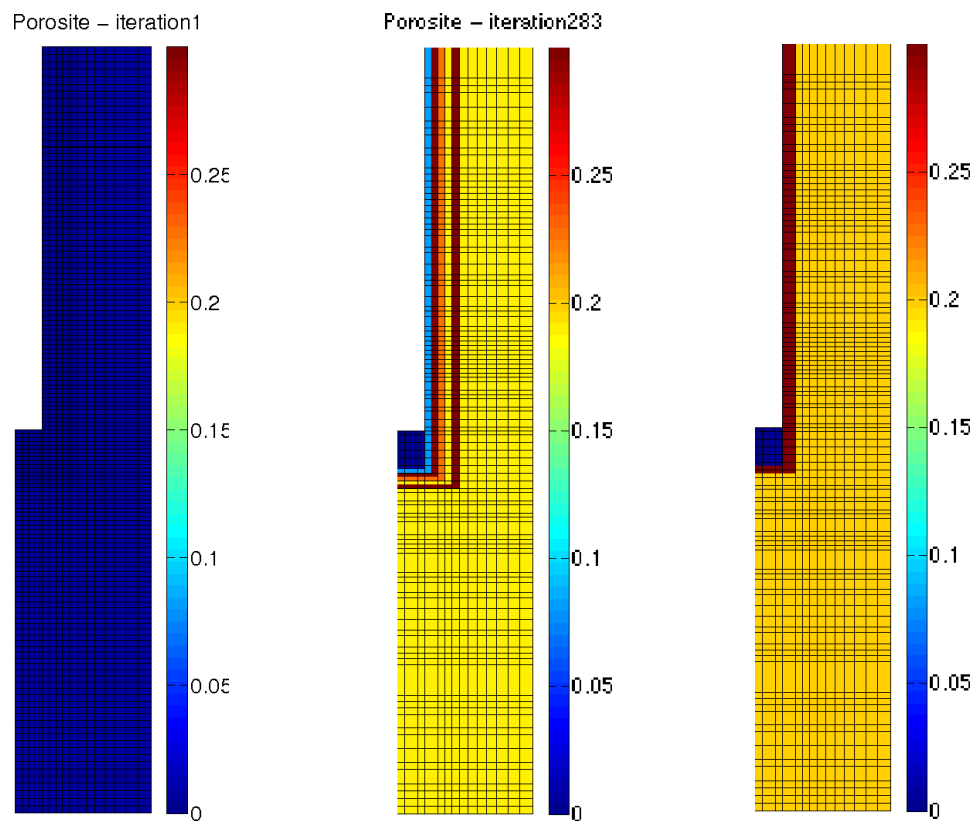


FIG. 6.22 – Porosité à l'itération 1, Porosité à l'itération 283, Porosité réelle

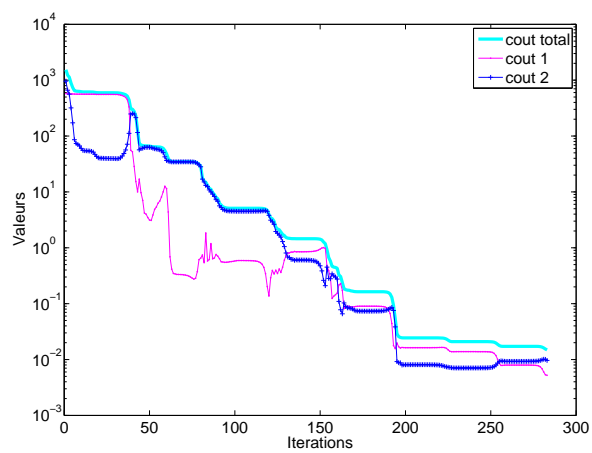


FIG. 6.23 – Evolution de la fonction coût

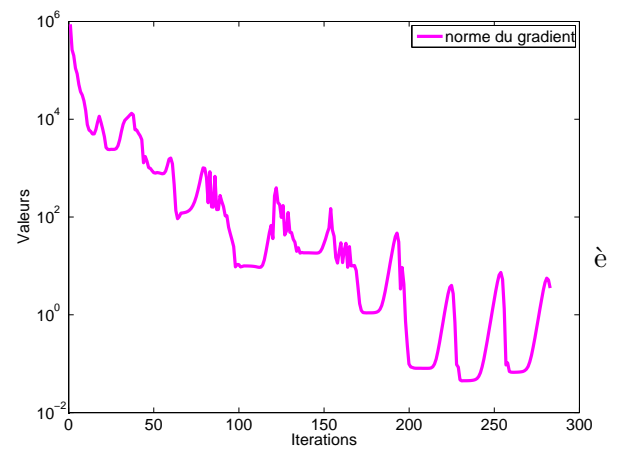


FIG. 6.24 – Evolution du gradient

Dans le cas d'un bruit de 5%, l'algorithme a itéré pendant 299 itérations et les résultats (tableau 6.7) montrent que la solution est toujours bien approchée pour la diffusion. Comme dans le cas du bruit de 3%, on observe encore à l'itération finale les trois zones recherchées (figure 6.25). Pour la porosité, on constate de nouveau que les coefficients ω_3 , ω_5 et ω_7 sont mal identifiés ce qui ne permet pas d'arriver à la configuration géologique exacte (figure 6.26). Néanmoins, comme dans le cas d'un bruit de 3%, une réponse peut être apportée à cet inconvénient (cf. section 6.3).

Les évolutions de la fonction coût et du gradient sont présentées sur les figures 6.27 et 6.28.

Diffusion	initiale	identifiés	réels	Porosité	initiale	identifiés	réels
D'_1	$1, 0.10^{-3}$	$9, 3847204159.10^{-2}$	0, 1	ω_1	$5, 0.10^{-3}$	0, 1814660222	0, 2
D'_3	$5, 0.10^{-3}$	0, 9232612745	1, 0	ω_3	$4, 0.10^{-3}$	$1, 0.10^{-3}$	0, 3
D'_4	$5, 0.10^{-3}$	0, 9487703310	1, 0	ω_4	$4, 0.10^{-3}$	0, 2764578564	0, 3
D'_5	$5, 0.10^{-3}$	$9, 8557550324.10^{-2}$	0, 1	ω_5	$4, 0.10^{-3}$	0, 3026078948	0, 2
D'_6	$5, 0.10^{-3}$	$9, 5695267349.10^{-2}$	0, 1	ω_6	$4, 0.10^{-3}$	0, 2038192260	0, 2
D'_7	$5, 0.10^{-3}$	$9, 6787129162.10^{-2}$	0, 1	ω_7	$4, 0.10^{-3}$	0, 3027834741	0, 2

TAB. 6.7 – Identification dans les zones 1, 3, 4, 5, 6 & 7 avec un bruit de 5%

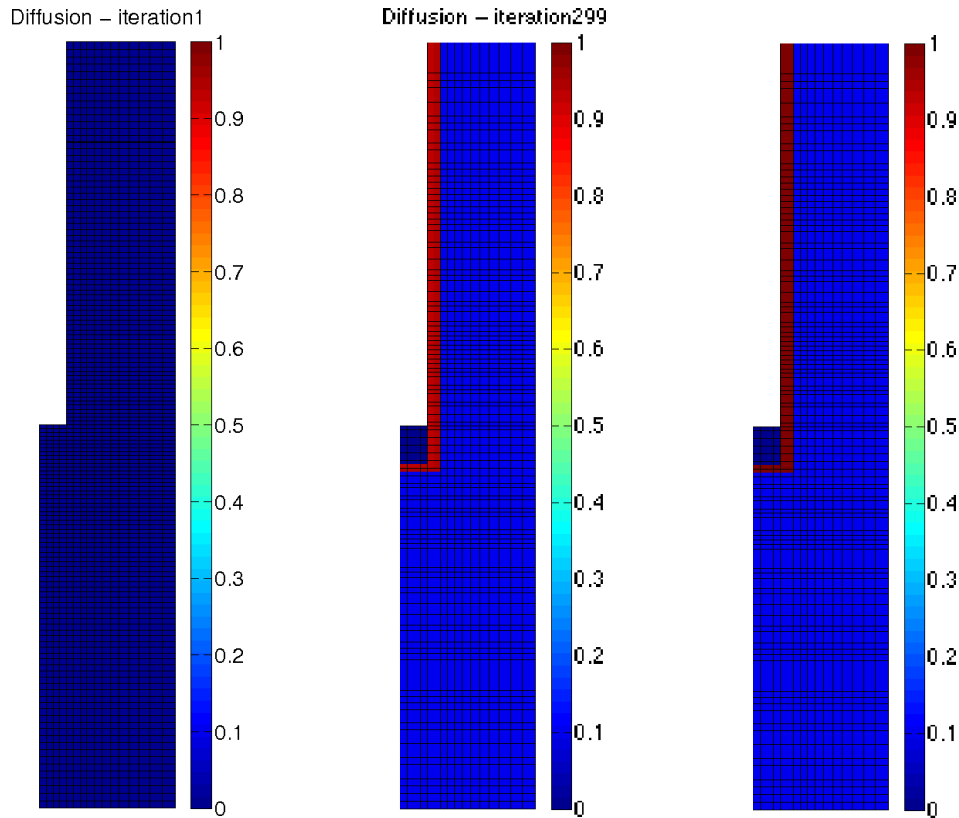


FIG. 6.25 – Diffusion à l'itération 1, Diffusion à l'itération 299, Diffusion réelle

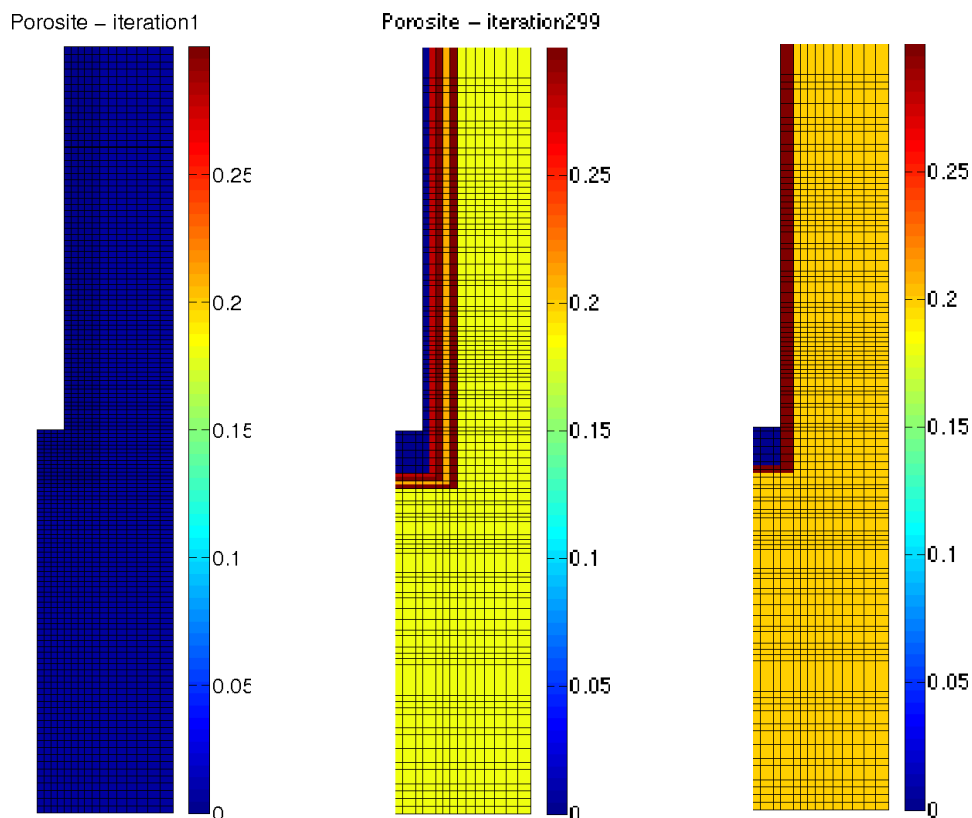


FIG. 6.26 – Porosité à l'itération 1, Porosité à l'itération 299, Porosité réelle

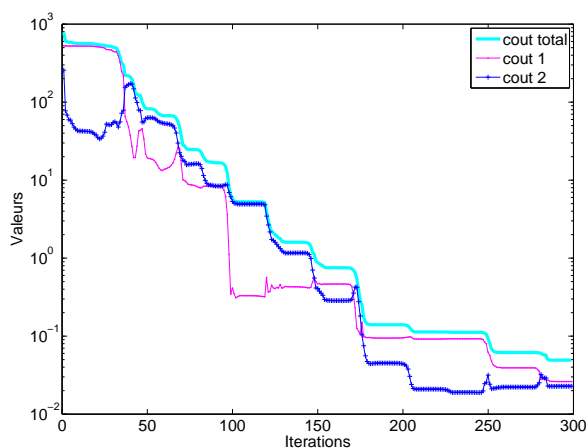


FIG. 6.27 – Evolution de la fonction coût

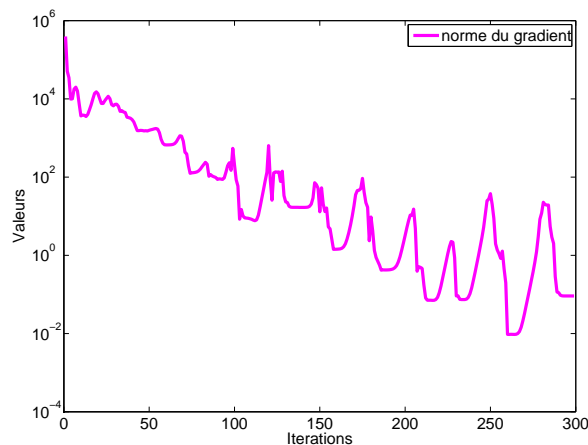


FIG. 6.28 – Evolution du gradient

Remarque 6.1 *On constate qu'avec un bruit pas trop élevé (\leq à 5%), l'algorithme reste très performant pour identifier les deux zones.*

Comme pour l'identification de paramètres, nous avons voulu voir, dans les deux tests qui suivent, les limites de l'algorithme en augmentant le bruit de façon significative.

Dans le cas d'un bruit de 10%, l'algorithme a itéré pendant 186 itérations et les résultats (tableau 6.8) montrent que la solution est encore satisfaisante pour la diffusion. On dis-

tingue à l'itération finale sur la figure 6.29 quatre zones dont deux sont très proches, on peut donc considérer (en regroupant ces deux zones) que l'on a bien retrouvé la configuration géologique réelle. Pour la porosité (tableau 6.9), bien que la plupart des coefficients soient proches des valeurs réelles (ω_1 , ω_5 , ω_6 et ω_7), on constate sur la figure 6.30 que l'on est pas arrivé à retrouver les zones recherchées.

L'évolution de la fonction coût et du gradient est présentée sur les figures 6.31 et 6.32.

Diffusion	initiale	identifiés	réels
D'_1	$5, 0.10^{-4}$	0, 1088427693	0, 1
D'_3	$1, 0.10^{-2}$	0, 9329235000	1, 0
D'_4	$1, 0.10^{-2}$	0, 8226169260	1, 0
D'_5	$1, 0.10^{-2}$	0, 1082165413	0, 1
D'_6	$1, 0.10^{-2}$	0, 1307609280	0, 1
D'_7	$1, 0.10^{-2}$	$9, 4241919088.10^{-2}$	0, 1

TAB. 6.8 – Identification de la diffusion avec un bruit de 10%

Porosité	initiale	identifiés	réels
ω_1	$5, 0.10^{-3}$	0, 2220867447	0, 2
ω_3	$2, 0.10^{-3}$	$9, 9999999999.10^{-5}$	0, 3
ω_4	$2, 0.10^{-3}$	$6, 9115608392.10^{-2}$	0, 3
ω_5	$2, 0.10^{-3}$	0, 1723855118	0, 2
ω_6	$2, 0.10^{-3}$	0, 2306095996	0, 2
ω_7	$2, 0.10^{-3}$	0, 2088383375	0, 2

TAB. 6.9 – Identification de la porosité avec un bruit de 10%

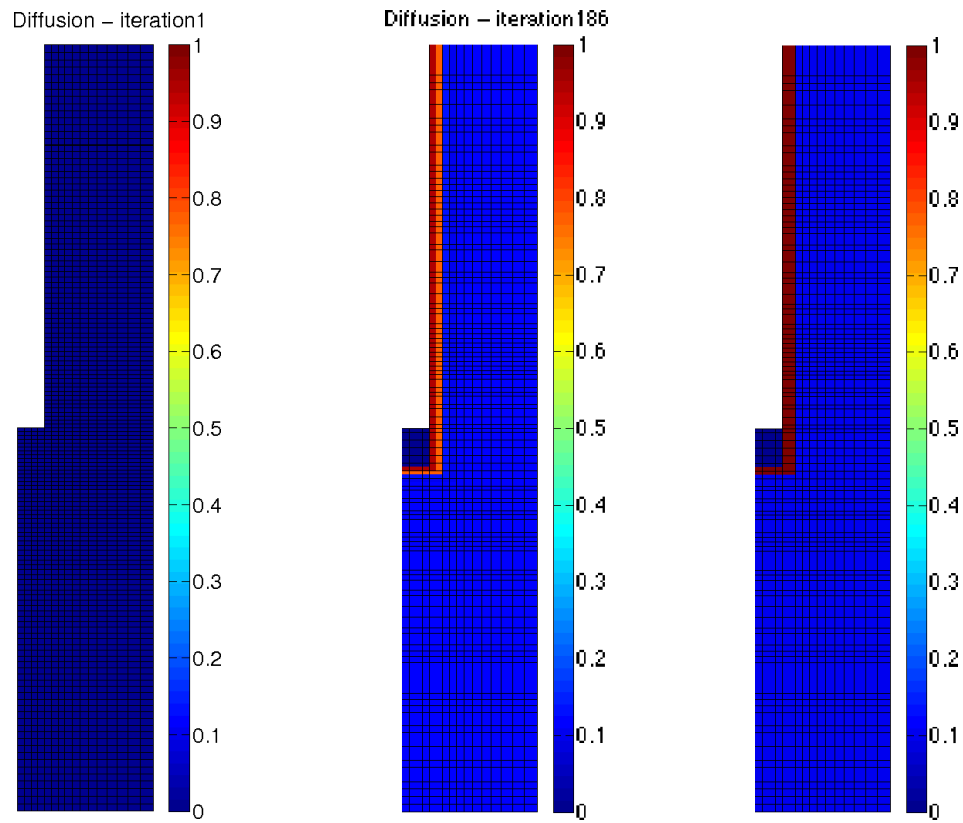


FIG. 6.29 – Diffusion à l'itération 1, Diffusion à l'itération 186, Diffusion réelle

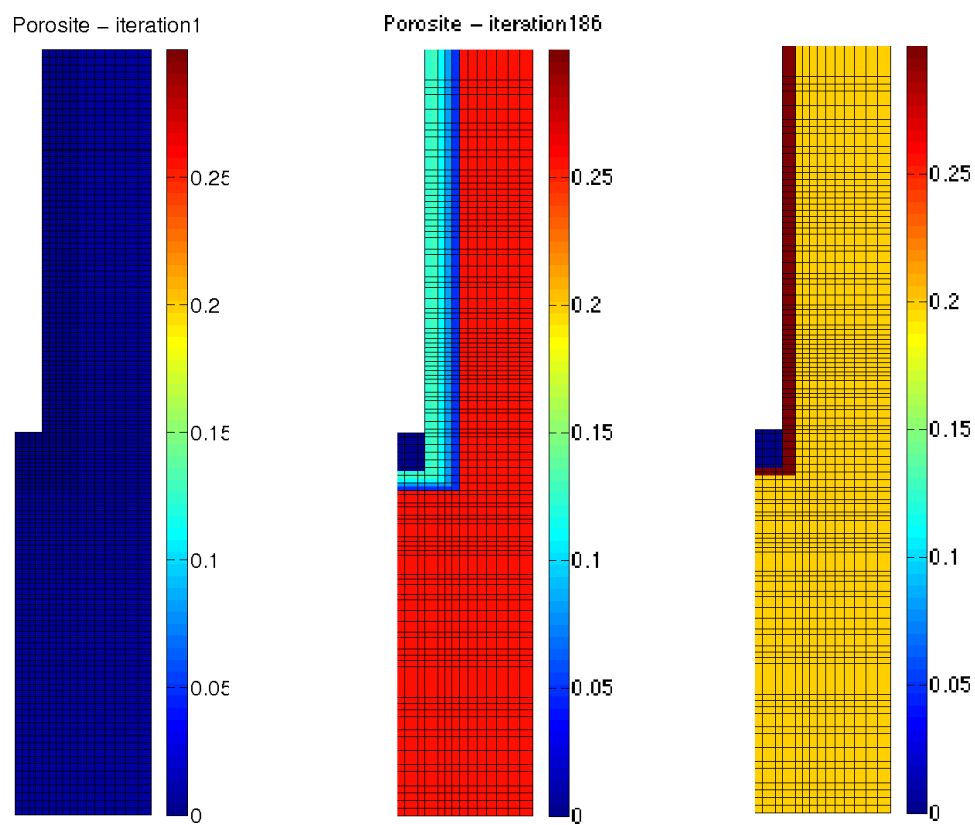


FIG. 6.30 – Porosité à l'itération 1, Porosité à l'itération 186, Porosité réelle

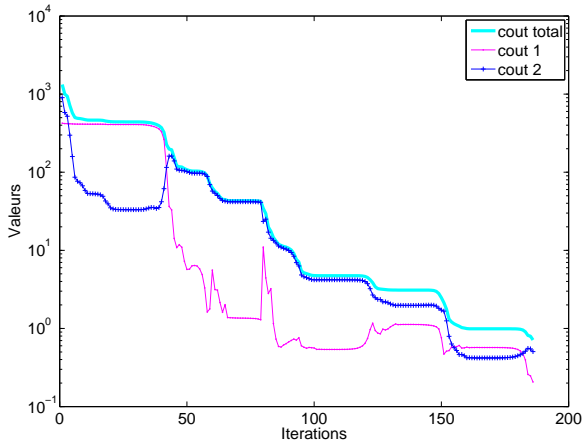


FIG. 6.31 – Evolution de la fonction coût

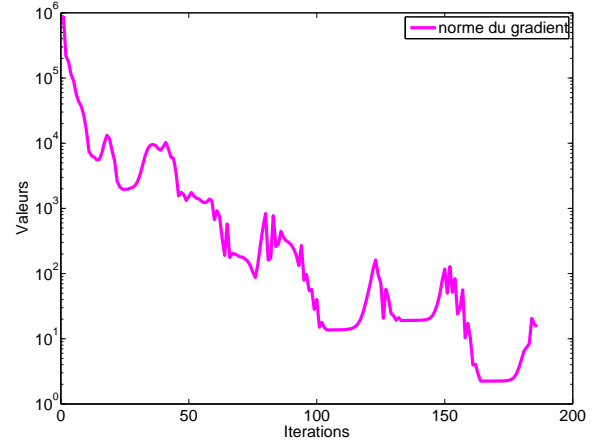


FIG. 6.32 – Evolution du gradient

Dans le cas d'un bruit de 20%, l'algorithme a itéré pendant 121 itérations et les résultats (tableau 6.10) montrent que seuls D'_1 et D'_7 ont été correctement identifiés. On voit sur la figure 6.33 qu'on est loin de la configuration exacte. Pour la porosité (tableau 6.11), on constate également que les coefficients identifiés ne sont pas bons et donc comme pour la diffusion, on a pas réussi à retrouver la configuration géologique exacte (figure 6.34). Ces résultats ne sont pas du tout surprenant dans la mesure où le bruit est très élevé.

L'évolution de la fonction coût et du gradient est présentée sur les figures 6.35 et 6.36.

Diffusion	initiale	identifiés	réels
D'_1	$5, 0.10^{-4}$	0, 1212964685	0, 1
D'_3	$1, 0.10^{-2}$	0, 5795504455	1, 0
D'_4	$1, 0.10^{-2}$	0, 6047805905	1, 0
D'_5	$1, 0.10^{-2}$	0, 3165127860	0, 1
D'_6	$1, 0.10^{-2}$	$8, 4095729074.10^{-2}$	0, 1
D'_7	$1, 0.10^{-2}$	0, 1027340394	0, 1

TAB. 6.10 – Identification de la diffusion avec un bruit de 20%

Porosité	initiale	identifiés	réels
ω_1	$5, 0.10^{-3}$	0, 2676178231	0, 2
ω_3	$2, 0.10^{-3}$	$7, 2659746084.10^{-3}$	0, 3
ω_4	$2, 0.10^{-3}$	$9, 0478753448.10^{-3}$	0, 3
ω_5	$2, 0.10^{-3}$	$9, 9999999999.10^{-5}$	0, 2
ω_6	$2, 0.10^{-3}$	$7, 2476646531.10^{-3}$	0, 2
ω_7	$2, 0.10^{-3}$	$1, 5985330457.10^{-2}$	0, 2

TAB. 6.11 – Identification de la porosité avec un bruit de 20%

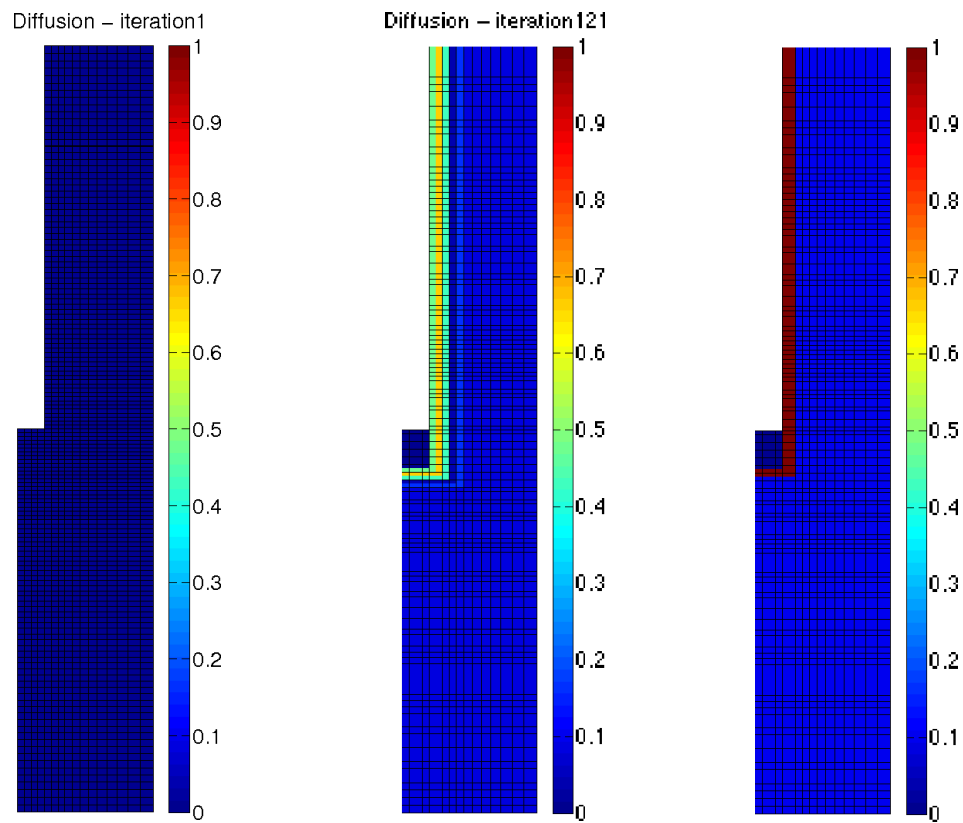


FIG. 6.33 – Diffusion à l'itération 1, Diffusion à l'itération 121, Diffusion réelle

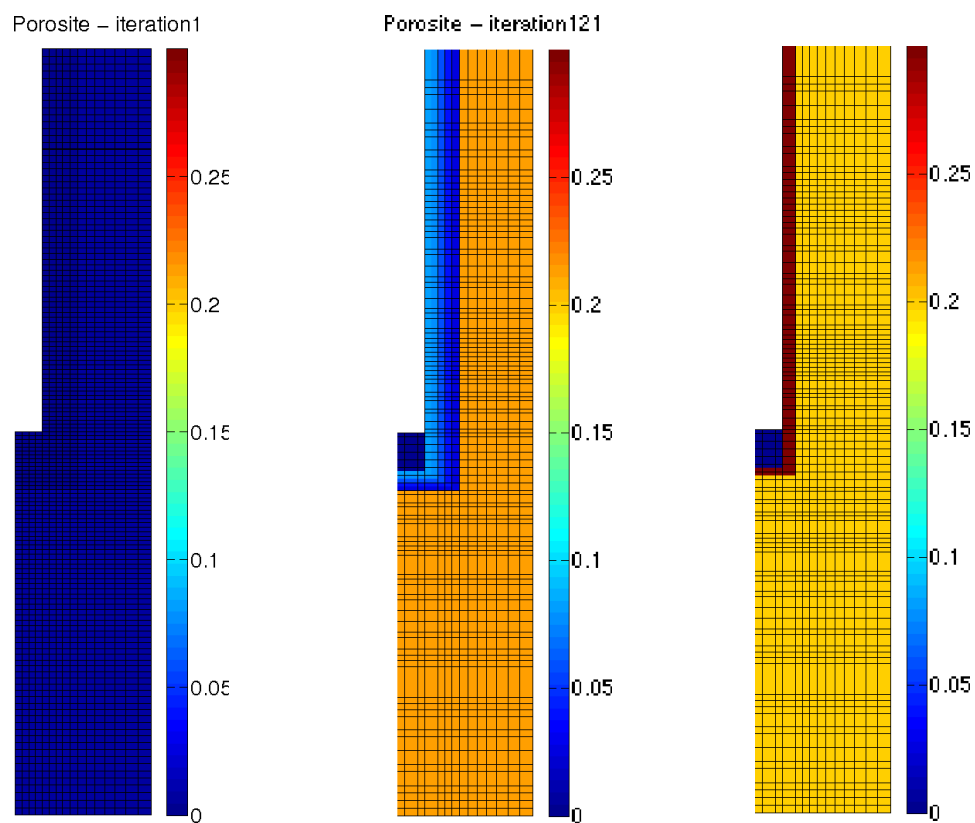


FIG. 6.34 – Porosité à l'itération 1, Porosité à l'itération 121, Porosité réelle

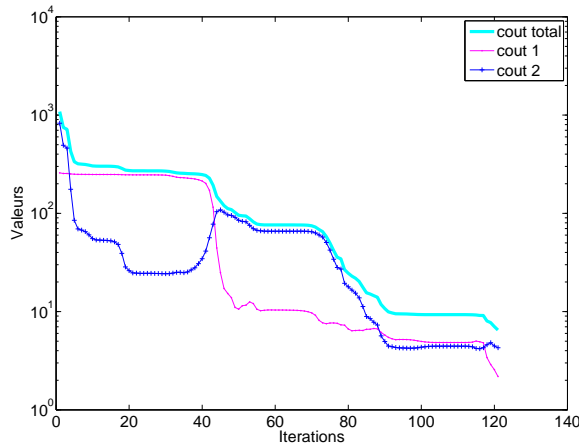


FIG. 6.35 – Evolution de la fonction coût

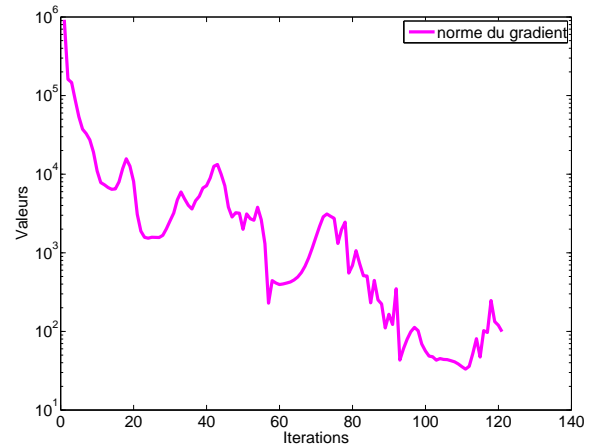


FIG. 6.36 – Evolution du gradient

6.3 Conclusion des tests

Nous avons appliqué dans ce chapitre la méthode d'identification de paramètres à notre problème d'identification de zones (cf. chapitre 1 et section 6.1). Les résultats obtenus (notamment sur le cas concentrique) ont montré la validité de la méthode employée. Ces tests ont également mis en évidence, tout comme pour l'estimation de paramètres, l'importance des observations (cf. section 6.2.1).

Les cas-tests ont été choisis en liaison avec les équipes de l'Andra (comme dans le chapitre 5) afin de représenter des problèmes physiques réels. Les premiers tests réalisés au début de ce chapitre ont montré que l'identification des paramètres par maille était possible mais que ce procédé était trop coûteux en mémoire et en temps. Une régularisation a été apportée en considérant des zones concentriques autour du forage (cf. section 6.2.2). On aurait également pu rajouter un terme régularisant au sens BV (pour conserver les discontinuités) ou au sens de la norme L^2 du gradient (cf. chapitre 3, section 3.2.3.3).

Les résultats dans le cas concentrique avec des données non bruitées se sont révélés très satisfaisants aussi bien avec des valeurs initiales proches des valeurs réelles qu'avec des valeurs initiales éloignées. En ce qui concerne l'identification avec données bruitées, les résultats obtenus sont bons pour la diffusion jusqu'à 10% ce qui nous a permis de retrouver la configuration géologique recherchée. Pour la porosité, nous nous sommes aperçus au cours des différents tests que nous n'arrivions pas à identifier correctement toutes les composantes de ce paramètre (et donc les zones). Cependant, il pourrait être possible de regrouper les zones en accord avec les valeurs de diffusion (qui est plus rapidement identifiée que la porosité), ce qui permettrait, partant de cette nouvelle configuration, d'identifier à nouveau la porosité et d'arriver à de meilleurs résultats.

Chapitre 7

Conclusion et perspectives

Nous nous sommes intéressés dans cette thèse à différents aspects de l'identification de paramètres en milieu poreux saturé. Dans un premier temps, nous avons étudié l'aspect identification de paramètres à zones fixées puis, dans un second temps, nous avons utilisé ce travail pour résoudre le problème d'identification de zones.

Le point de départ de cette thèse répond à l'expression de besoins de l'Andra pour la recherche de méthodes numériques et le développement d'outils sur le thème de l'estimation de paramètres dans des couches géologiques profondes puisque les expériences en laboratoire ou en milieu naturel menées ne permettent pas d'évaluer les coefficients de diffusion et de porosité directement. L'objectif a été de mettre au point la théorie mathématique pour résoudre le problème d'identification de ces paramètres puis de retrouver les zones dont ces coefficients sont le support. La partie théorique a ensuite été implémentée informatiquement et plusieurs tests, préalablement choisis en liaison avec les équipes de l'Andra, ont été effectués afin de pouvoir apprécier la méthode conçue et développée. L'interprétation physique des résultats obtenus a été primordiale et a constitué le fil conducteur de ce projet.

L'idée a été de créer un outil flexible qui puisse s'appliquer aussi bien aux essais sur échantillons (à une échelle d'espace de l'ordre du centimètre) qu'aux mesures effectuées dans l'expérience DIR (Diffusion de traceurs Inertes et Réactifs) au laboratoire de Bure (pour laquelle l'ordre de grandeur est le mètre) ou encore aux mesures en milieu naturel dans un cadre géologique sur des échelles de la centaine de mètres. Dans le prolongement des travaux réalisés au sein du thème Problèmes inverses et analyse de sensibilité du GNR MoMaS, nous avons utilisé la technique de zonation qui consiste à décomposer le milieu en zones homogènes dans lesquelles les coefficients ont été identifiés. Pour cela, nous avons travaillé au développement du code adjoint du code de transfert par diffusion-dispersion-convection Traces en collaboration avec l'équipe Tropics de l'INRIA Sophia-Antipolis. La zonation, dans le cas-test "réel" (proche de l'expérience DIR), a pris en compte la zone endommagée par le forage. Le calage de l'outil a d'abord été effectué à zones fixées, puis, dans un second temps, l'identification de zones a été réalisée.

On s'est limité à travailler sur des problèmes 2D (au lieu du 2D axisymétrique), cependant cela ne remet pas en question la démarche suivie mais nous a permis de gagner en temps calcul par rapport au 3D. De plus, l'étape de résolution du problème en 2D était quasiment nécessaire (et plus facile à mettre œuvre en 2D qu'en 3D) pour vérifier et valider la méthode avant de passer au 2D axisymétrique ou au 3D. Nous avons eu aussi bien dans le cas de l'estimation de paramètres que dans l'identification des zones des résultats très satisfaisants avec des données bruitées et non bruitées, ce qui est encourageant pour la

résolution en 3D.

L'analyse de sensibilité du résultat de l'identification de paramètres par rapport aux erreurs sur les mesures telle qu'exposée dans [67] n'a pas été traitée car elle nécessitait le calcul de l'adjoint au second ordre qui est assez lourd. Si l'analyse de sensibilité montre que l'erreur sur les résultats est du même ordre que l'incertitude sur les mesures alors la méthode d'identification donnera satisfaction.

Concernant l'identification de zones, une perspective de travail (après celle évoquée dans la conclusion du chapitre 6) serait de mettre au point une méthode itérative permettant de modifier les zones à chaque étape et qui consisterait à

- identifier les coefficients dans toutes les zones,
- regrouper des zones par coefficients proches,
- répéter le processus en identifiant les coefficients dans les nouvelles zones (où il y en a moins).

On pourrait également tester la méthode des indicateurs de raffinement et la méthode d'optimisation topologique. Il serait en effet intéressant, lors d'un futur travail, de comparer la précision et l'efficacité de notre méthode avec ces deux techniques, d'autant plus qu'aucune des deux n'a encore été employée pour résoudre ce type de problème.

Ce travail nous encourage à continuer les tests numériques sur le modèle (1.3) en testant le cas convectif (vitesse de Darcy non nulle) avec un terme de filiation. On pourrait aussi imaginer rajouter une zone à la configuration du cas-test "réel" en divisant la zone endommagée en deux : une zone pour la partie fracturée et une zone pour la partie fissurée. L'objectif principal reste clairement à court terme de passer à la géométrie 3D du cas-test "réel" et d'utiliser, dès qu'elles seront disponibles, les données réelles issues de l'expérimentation DIR. L'utilisation des données réelles est d'autant plus importante qu'on sera en mesure, en confrontant les résultats à l'expérience, de valider (resp. invalider) la représentation du modèle qu'on s'est fixé.

Annexe A

Nomenclature des grandeurs physiques

Symbole	Description	Dimension
ω	Porosité	Adimensionné
R	Facteur de retard	Adimensionné
h	Charge hydraulique	L
p	Pression	$M.L^{-1}.T^{-2}$
g	Gravité	$M.T^{-2}$
μ	Viscosité du fluide	$M.L^{-1}.T^{-1}$
ρ	Densité du fluide	$M.L^{-3}$
q	Vitesse de Darcy	$L.T^{-1}$
C	Concentration	$M.L^{-3}$
s	Coefficient de stockage	L^{-1}
S	Terme source	$M.L^{-3}.T^{-1}$
k	Perméabilité	L^2
K	Conductivité hydraulique	$L.T^{-1}$
α_L	Dispersivité longitudinale	L
α_T	Dispersivité transversale	L
D_e	Coefficient de diffusion effectif	$L^2.T^{-1}$
D_m	Coefficient de diffusion moléculaire	$L^2.T^{-1}$
λ	Coefficient de décroissance radioactive	T^{-1}
ϱ	Constante cinétique de précipitation/dissolution	T^{-1}

TAB. A.1 – Valeurs des grandeurs physiques

Bibliographie

- [1] ADOL-C. COmputational INfrastructure for Operations Research (COIN-OR). <http://www.coin-or.org/projects/ADOL-C.xml>, 2008.
- [2] C. Alboin, J. Jaffré, J. E. Roberts, X. Wang, and C. Serres. Domain decomposition for some transmission problems in flow in porous media. In *Numerical treatment of multiphase flows in porous media (Beijing, 1999)*, volume 552 of *Lecture Notes in Phys.*, pages 22–34. Springer, Berlin, 2000.
- [3] C. Alboin, J. Jaffré, J. E. Roberts, X. Wang, and C. Serres. Modeling fractures as interfaces for flow and transport in porous media. In *Fluid flow and transport in porous media : mathematical and numerical treatment (South Hadley, MA, 2001)*, volume 295 of *Contemp. Math.*, pages 13–24. Amer. Math. Soc., Providence, RI, 2002.
- [4] G. Allaire and F. Jouve. A level-set method for vibration and multiple loads structural optimization. *Comput. Methods Appl. Mech. Engrg.*, 194(30-33) :3269–3290, 2005.
- [5] G. Allaire and F. Jouve. Minimum stress optimal design with the level set method. *Engineering Analysis with Boundary Elements*, 32 :909–918 (2008). Internal report, n. 618, CMAP, Ecole Polytechnique (June 2007), June, 2007.
- [6] G. Allaire, F. Jouve, and A.-M. Toader. Structural optimization using sensitivity analysis and a level-set method. *J. Comp. Phys.*, 194 :363–393, 2004.
- [7] S. Amstutz, I. Horchani, and M. Masmoudi. Crack detection by the topological gradient method. *Control Cybernet.*, 34(1) :81–101, 2005.
- [8] Andra. *Dossier Argile 2005*. Rapport Andra, 2005.
- [9] J.-B. Apoung, P. Havé, J. Houot, M. Kern, and A. Semin. Reactive transport in porous media. *ESIAM Proceedings, submitted*, 2008.
- [10] D. N. Arnold and F. Brezzi. Mixed and nonconforming finite element methods : implementation, postprocessing and error estimates. *RAIRO Modél. Math. Anal. Numér.*, 19(1) :7–32, 1985.
- [11] D. N. Arnold, F. Brezzi, B. Cockburn, and D. Marini. Discontinuous Galerkin methods for elliptic problems. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 89–101. Springer, Berlin, 2000.
- [12] D. Auroux and J. Blum. Data assimilation methods for an oceanographic problem in multidisciplinary methods for analysis, optimization and control of complex systems. *Mathematics in Industry*, 6 :179–194, 2004.
- [13] D. Auroux and M. Masmoudi. Image processing by topological asymptotic expansion. *J. Math. Imaging Vision*, 33(2) :122–134, 2009.
- [14] L. J. Belaid, M. Jaoua, M. Masmoudi, and L. Siala. Image restoration and edge detection by topological asymptotic expansion. *C. R. Math. Acad. Sci. Paris*, 342(5) :313–318, 2006.

- [15] H. Ben Ameur, G. Chavent, and J. Jaffré. Refinement and coarsening indicators for adaptive parametrization : application to the estimation of hydraulic transmittivities. *Inverse Problems*, 20 :673–696, 2004.
- [16] H. Ben Ameur, F. Clément, P. Weis, and G. Chavent. The multidimensional refinement indicators algorithm for optimal parameterization. *J. Inverse Ill-Posed Probl.*, 16(2) :107–126, 2008.
- [17] H. Ben Ameur and B. Kaltenbacher. Regularization of parameter estimation by adaptive discretization using refinement and coarsening indicators. *J. Inverse Ill-Posed Probl.*, 10(6) :561–583, 2002.
- [18] M. P. Bendsøe. Optimal topology design of continuum structure : an introduction. *Technical report, Department of Mathematics, Technical University of Denmark*, 1996.
- [19] V. Berinde. *Iterative approximation of fixed points*, volume 1912 of *Lecture Notes in Mathematics*. Springer, Berlin, second edition, 2007.
- [20] J. Blum. *Numerical simulation and optimal control in plasma physics*. Wiley/Gauthier-Villars Series in Modern Applied Mathematics, Chichester, 1989.
- [21] J. Blum. Problèmes inverses : de l’expérimentation à la modélisation. *OFTA, ARAGO*, 1999.
- [22] J. Blum, G. Dobranszky, R. Eymard, and R. Masson. Inversion of a stratigraphic model with seismic constraints. *Under revision for Inverse Problems*, 2006.
- [23] J.F. Bonnans, J.C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal. *Numerical optimization*. Universitext. Springer-Verlag, Berlin, 2003. Theoretical and practical aspects, Translated and revised from the 1997 French original.
- [24] F. Brezzi and M. Fortin. *Mixed and hybrid finite element methods*, volume 15 of *Springer Series in Computational Mathematics*. Springer-Verlag, New York, 1991.
- [25] F. Brezzi, D. Marini, P. Pietra, and A. Russo. Discontinuous finite elements for diffusion problems. *Numerical Methods for Partial Differential Equations*, 1999, submitted.
- [26] S. Buschaert and al. The contribution of water geochemistry to the understanding of the regional hydrogeological system. *Bull. Soc. Geol. de France.*, 2006.
- [27] Dan G. Cacuci. Sensitivity theory for nonlinear systems. *J. Math. Phys.*, 22(12) :2794–2812, 1981.
- [28] J. Carrera. State of the art of the inverse problem applied to the flow and solute transport equation. *Groundwater Flow and Quality Modeling*, NATO-ASI Series C :549–583, 1988.
- [29] A. Cartalade and al. Application of inverse modeling methods to thermal and diffusion experiments at mont terri rock laboratory. *Proc. of the 2nd international meeting : Clays in Natural and Engineered Barriers for Radioactive waste confinement, Tours*, 2005.
- [30] A. Cartalade, P. Montarnal, B. Cavanna, and J. Blum. Estimation des paramètres de transport d’un milieu poreux, approche par état adjoint. *Rapport DM2S/SFME/MTMS/RT/03-002/A. CEA/Saclay*, 2003.
- [31] J. Céa, S. Garreau, P. Guillaume, and M. Masmoudi. The shape and topological optimizations connection. *Comput. Methods Appl. Mech. Engrg.*, 188(4) :713–726, 2000. IV WCCM, Part II (Buenos Aires, 1998).
- [32] G. Chavent. Identification of functional parameters in partial differential equations. In *Identification of parameters in distributed systems (Joint Automat. Control Conf., Univ. Texas, Austin, Tex., 1974)*, pages 31–48. Amer. Soc. Mech., Engrs., New York, 1974.

- [33] G. Chavent. Identification of distributed parameter systems : about the output least square method, its implementation and identifiability. *Proc. of the 5th IFAC Symposium on Identification and System Parameter Estimation*, pages 85–97, 1979.
- [34] G. Chavent. On the theory of practice of nonlinear least-squares. *Adv. in Water Res.*, 14(2) :55–63, 1991. Parameter identification in ground water flow, transport, and related processes, Part I.
- [35] G. Chavent. Strategies for the regularization of nonlinear least squares problems. In *Inverse problems in diffusion processes (Lake St. Wolfgang, 1994)*, pages 217–232. SIAM, Philadelphia, PA, 1995.
- [36] G. Chavent and R. Bissel. Indicator for the refinement of parametrization. In *Proceedings of the International Symposium on Inverse Problems in Engineering Mechanics*, pages 185–190. Nagano, Japan, March, 1998.
- [37] G. Chavent, J. Jaffré, and S. Jan-Jégou. Estimation of relative permeabilities in three-phase flow in porous media. *Inverse Problems*, 15(1) :33–39, 1999. Conference on Inverse Problems, Control and Shape Optimization (Carthage, 1998).
- [38] G. Chavent, J. Jaffré, and J. E. Roberts. Mixed-hybrid finite elements and cell-centered finite volumes for two-phase flow in porous media. In A. Bourgeat, C. Carasso, S. Luckhaus, and A. Mikelic, editors, *Mathematical Modelling of Flow Through Porous Media*, pages 100–114, World Scientific, Singapore, 1995.
- [39] G. Chavent and J. Liu. Multiscale parametrization for the estimation of a diffusion coefficient in elliptic and parabolic problems. *5th IFAC Symposium on Control of Distributed Parameter Systems*, A. El Jai and M. Amouroux eds., Université de Perpignan, France :315–324, June 26-29, 1989.
- [40] P.G. Ciarlet. *Introduction à l'analyse numérique matricielle et à l'optimisation*. Dunod, 2006.
- [41] F. Clément, N. Khvoenkova, A. Cartalade, and P. Montarnal. Analyse de sensibilité et estimation de paramètres de transport pour une équation de diffusion, approche par état adjoint. *INRIA*, Rapport n°5132, 2004.
- [42] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin methods (Newport, RI, 1999)*, volume 11 of *Lect. Notes Comput. Sci. Eng.*, pages 3–50. Springer, Berlin, 2000.
- [43] P. Courtier and O. Talagrand. Variational assimilation of meteorological observations with the adjoint equation – Part II. Numerical results. *Q. J. R. Meteorol. Soc.*, 113 :1329 – 1347, 1987.
- [44] C. de Dieuveult, J. Erhel, and M. Kern. A global strategy for solving reactive transport equations. *J. Computational Physics*, submitted, 2008.
- [45] G. De Marsily and al. Regards sur 40 ans de problèmes inverses en hydrogéologie. *CRAS, Sciences de la terre et des plantes*, 329 :73–87, 1999.
- [46] C. Delannoy. *Programmer en Fortran 90*. Eyrolles, Paris, 2004.
- [47] S. Dewonck. Expérimentation DIR, synthèse des résultats obtenus sur les forages EST208, DIR2002 et DIR2003. *Andra, Document interne*, DP/EST/SS, 2005.
- [48] H. W. Engl, M. Hanke, and A. Neubauer. *Regularization of inverse problems*, volume 375 of *Mathematics and its Applications*. Kluwer Academic Publishers Group, Dordrecht, 1996.
- [49] M. Farhloul and A. Serghini Mounim. A mixed-hybrid finite element method for convection-diffusion problems. *Appl. Math. Comput.*, 171(2) :1037–1047, 2005.

- [50] P.-A. Faure. Gmshwrapper. Encapsulation de Gmsh en Python. *Rapport de stage de Master, Andra & Université Claude Bernard Lyon 1*, 2006.
- [51] S. Garreau, Ph. Guillaume, and M. Masmoudi. The topological asymptotic for PDE systems : the elasticity case. *SIAM J. Control Optim.*, 39(6) :1756–1778 (electronic), 2001.
- [52] J.C Gilbert, G. Le Vey, and Masse J. La différentiation automatique de fonctions représentées par des programmes. *INRIA*, Rapport n°1557, 1991.
- [53] J.C. Gilbert and C. Lemaréchal. Some numerical experiments with variable-storage quasi-Newton algorithms. *Math. Programming*, 45(3, (Ser. B)) :407–435, 1989.
- [54] J.C. Gilbert and C. Lemaréchal. The module M1QN3. Version 3.1. *INRIA Rocquencourt & Rhône-Alpes*, 2006.
- [55] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and Software*, pages 1, 35–54, 1992.
- [56] A. Griewank. Some bounds on the complexity of gradients, Jacobians, and Hessians. In *Complexity in numerical optimization*, pages 128–162. World Sci. Publ., River Edge, NJ, 1993.
- [57] A. Griewank. Evaluating derivatives. principles and techniques of algorithmic differentiation. *Frontiers in applied mathematics*, SIAM(1) :35–54, 2000.
- [58] Ph. Guillaume and K. Sid Idris. The topological asymptotic expansion for the Dirichlet problem. *SIAM J. Control Optim.*, 41(4) :1042–1072 (electronic), 2002.
- [59] L. Hascoet. Analyses statiques et transformations de programmes : de la parallélisation à la différentiation. *Mémoire d’habilitation à diriger des recherches. Université de Nice-Sophia-Antipolis*, 2005.
- [60] M. Hayek, Ph. Ackerer, and E. Sonnendrücker. A new refinement indicator for adaptive parameterization : application to the estimation of the diffusion coefficient in an elliptic problem. *J. Comput. Appl. Math.*, 224(1) :307–319, 2009.
- [61] J.-B. Hiriart-Urruty and C. Lemaréchal. *Convex analysis and minimization algorithms. I & II*, volume 305–306 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 1993. Fundamentals.
- [62] H. Hoteit and P. Ackerer. *Traces : User’s Guide V 1.00*. ULP, ANDRA, CNRS. Fluid and Solid Mechanics Institute, Strasbourg, 2003.
- [63] J. Jaffré. Flux calculation at the interface between two rock types for two-phase flow in porous media. *Transport in Porous Media*, preprint in report *INRIA n° 2075*, pages 21 :195–207, 1995.
- [64] J. Jaffré. Numerical calculation of the flux across an interface between two rock types of a porous medium for a two-phase flow. In *Hyperbolic problems : theory, numerics, applications (Stony Brook, NY, 1994)*, pages 165–177. World Sci. Publ., River Edge, NJ, 1996.
- [65] M. Kern. *Problèmes inverses*. Techniques de l’Ingénieur, to appear, TSI, 2008.
- [66] F.-X. Le Dimet and J. Blum. Assimilation de données pour les fluides géophysiques. *Matapli n° 67*, 2002.
- [67] F.-X. Le Dimet, I.M. Navon, and D.N. Daescu. Second order information in data assimilation. *Monthly Weather Review*, 2002.

- [68] F.-X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus*, 38A :97–110, 1986.
- [69] C. Lemaréchal and E. Panier. Les modules M2QN1 et MQHESS. Version 2.0a. *INRIA Rhône-Alpes*, 2000.
- [70] J.-L. Lions. *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*. Avant propos de P. Lelong. Dunod, Paris, 1968.
- [71] J. Liu. A multiresolution method for distributed parameter estimation. *SIAM J. Sci. Comput.*, 14(2) :389–405, 1993.
- [72] B. Luong, J. Blum, and J. Verron. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems, IOP*, 14(4) :979–997, 1998.
- [73] E. Marchand. Analyse de sensibilité déterministe pour la simulation numérique du transfert de contaminants. *Thèse de doctorat, Université Paris IX Dauphine*, 2007.
- [74] E. Marchand, F. Clément, J.E. Roberts, and G. Pépin. Deterministic sensitivity analysis for a model for flow in porous media. *Adv. in Water Res.*, 31 :1025–1037, 2008.
- [75] A. Martelli. *Python en concentré*. O’Reilly, Paris, 2007.
- [76] V. Martin. Simulations multidomaines des écoulements en milieu poreux. *Thèse de doctorat, Université Paris IX Dauphine*, 2004.
- [77] D. Mc Laughlin and L.R. Townley. A reassessment of the groundwater inverse problem. *Water Resources Research.*, 32 (5) :1131–1161, 1996.
- [78] U. Meissner. A mixte finite element model for use in potential flow problem. *Int. J. Numer. Methods Eng.*, 6 :467–473, 1973.
- [79] J.-C. Nédélec. A new family of mixed finite elements in \mathbf{R}^3 . *Numer. Math.*, 50(1) :57–81, 1986.
- [80] S. P. Neuman and J. Carrera. Maximum-likelihood adjoint-state finite-element estimation of groundwater parameters under steady- and non-steady-state conditions. *System identification and parameter estimation of environmental systems, Appl. Math. Comput.*, 17(4) :405–432, 1985.
- [81] H. Nifenecker and G. Ouzounian. Le stockage des déchets nucléaires en site profond. *EDP Sciences*, 2006.
- [82] R.E. Plessix, G. Chavent, and Y.H. De Roeck. A quantitative migration to estimate the 2D velocity distribution. *Mathematical and Numerical aspects of Wave propagation, SIAM*, 1995.
- [83] P.-A. Raviart and J. M. Thomas. A mixed finite element method for 2nd order elliptic problems. In *Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (C.N.R.), Rome, 1975)*, pages 292–315. Lecture Notes in Math., Vol. 606. Springer, Berlin, 1977.
- [84] W. H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Tech. Report LA-UR-73-479*, Los Alamos Scientific Laboratory, 1973.
- [85] J. E. Roberts and J.-M. Thomas. Mixed and hybrid finite element methods. *INRIA-Rocquencourt, Rapport n°737*, 1987.
- [86] D. Savage. *The Scientific and Regulatory Basis for the Geological Disposal of Radioactive Waste*. Wiley, 1995.
- [87] P. Siegel, R. Mosé, Ph. Ackerer, P. Blaschke, and J. Jaffré. Solution of advection dispersion equation using operator and time splitting. In *Proceedings of IX International Conference on Finite Elements in Fluids, New Trends and Applications, Venice, Italy (October 15-21)*, 1995. ARMOIRE.

- [88] P. Siegel, R. Mosé, Ph. Ackerer, and J. Jaffré. Solution of the advection-diffusion equation using a combination of discontinuous and mixed finite elements. *Internat. J. Numer. Methods Fluids*, 24(6) :595–613, 1997.
- [89] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM Rev.*, 40(1) :110–112 (electronic), 1998.
- [90] N.-Z. Sun. *Inverse Problems in Groundwater Modeling*. Kluwer Academic Publishers, the Netherland, 1995.
- [91] TAMC. <http://www.autodiff.com/tamc/>. *Tangent linear and Adjoint Model Compiler*, 2008.
- [92] Tapenade. <http://www-sop.inria.fr/tropics/>. *Tropics Team, INRIA Sophia-Antipolis*, 2008.
- [93] A. Tarantola. *Inverse problem theory : Methods for data fitting and model parameter estimation*. Elsevier Science Publishers B.V., Amsterdam, 1987.
- [94] R. Temam and J. Tribbia, editors. *Handbook of numerical analysis. Vol. XIV. Special volume : computational methods for the atmosphere and the oceans*, volume 14 of *Handbook of Numerical Analysis*. Elsevier/North-Holland, Amsterdam, 2009.
- [95] J.M. Thomas. Sur l'analyse numérique des méthodes d'éléments finis hydrides et mixtes. *Thèse de doctorat, Université Pierre et Marie Curie*, 1977.
- [96] V. Thomée. *Galerkin finite element methods for parabolic problems*, volume 25 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 2006.
- [97] A. Tikhonov and V. Arsenin. *Solutions of ill-posed problems*. V. H. Winston & Sons, Washington, D.C. : John Wiley & Sons, New York, 1977. Translated from the Russian, Preface by translation editor Fritz John, Scripta Series in Mathematics.
- [98] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Engrg.*, 192(1-2) :227–246, 2003.
- [99] D.A. Zimmermann and al. A comparison of seven geostatistically based inverse approaches to estimate transmissivities for modeling advective transport by groundwater flow. *Water Resources Research*, 34, n°6 :1373–1413, 1998.

Résumé

Dans ce travail, nous nous sommes intéressés à l'estimation des paramètres de transport de solutés dans un milieu poreux peu perméable, d'un point de vue théorique et appliqué, en lien avec le stockage des déchets nucléaires à vie longue en couche géologique profonde. La problématique de la gestion des déchets nucléaires est devenue durant ces dernières années une préoccupation importante de la société. Les modèles mathématiques utilisés pour le stockage des déchets en site profond sont basés sur les équations d'écoulement et de transport en milieu poreux. Ces modèles nécessitent de connaître les paramètres physiques des différentes couches géologiques. Ces paramètres (porosité et diffusion) n'étant pas accessibles directement, nous avons étudié, dans un premier temps, l'aspect identification de paramètres à zones fixées. Pour résoudre notre problème inverse, nous avons utilisé des méthodes déterministes (ou variationnelles) basées sur la minimisation d'un critère (ou fonction coût) quadratique, l'objectif étant de trouver l'ensemble des paramètres qui minimisent le critère. Nous avons décomposé le milieu en zones homogènes dans lesquelles les coefficients ont été identifiés (technique de zonation). Pour cela, nous avons différencié le code de convection-diffusion Traces avec le logiciel de différentiation automatique Tapenade afin d'obtenir le code adjoint de Traces. Dans un second temps, nous avons utilisé ce travail pour résoudre le problème d'identification de zones. On a cherché, partant d'une configuration géologique modifiée, à retrouver la configuration géologique réelle en identifiant les paramètres qui ont pour support ces zones. En regroupant ensuite par isovaleurs les coefficients identifiés, nous avons obtenu la configuration recherchée.

Mots-clés : déchets nucléaires, équations d'écoulement et de transport, porosité, diffusion, milieu hydro-géologique, problème inverse, état adjoint, estimation de paramètres, zonation, méthodes numériques, Traces, Tapenade.

Summary

In this work, we are interested in estimating solute transport parameters in a porous medium with a low permeability, from a theoretical and applied point of view in connection with the storage of long-lived nuclear waste in deep geological layer. In recent years, the problem of nuclear waste management has become a major concern of society. The mathematical models used for storing waste in deep level are based on the flow and transport equations in porous media. These models require the knowledge of the physical parameters of the different geological layers. Since those parameters (porosity and diffusion) are not directly accessible by measurements, we studied firstly the aspect of parameters identification in fixed areas. To solve this inverse problem, we used deterministic (or variational) methods based on the minimization of a quadratic criterion (or cost function) with the objective of finding all the parameters that minimize the criterion. We have broken down the media in homogeneous areas in which the coefficients have been identified (zonation technique). For this, we differentiated the convection-diffusion code Traces with the automatic differentiation software Tapenade to get the adjoint code of Traces. In a second step, we used this work to solve the zones identification problem. Starting from a modified geological configuration, we searched for the actual geological configuration by identifying the parameters whose supports are these zones. Then, by grouping the identified coefficients by isovalues, we obtained the real configuration.

Keywords : nuclear waste, flow and transport equations, porosity, diffusion, hydro-geological media, inverse problem, adjoint state, parameter estimation, zonation, numerical methods, Traces, Tapenade.