

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (Toulouse INP)

Discipline ou spécialité :

Mathématiques Appliquées

Présentée et soutenue par :

M. ANTOINE BERNIGAUD

le vendredi 16 décembre 2022

Titre :

Introduction de la norme Lp pour la prise en compte de propriétés de parcimonie en assimilation de données.

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Institut de Recherche en Informatique de Toulouse (IRIT)

Directeur de Thèse :

M. SERGE GRATTON

Rapporteurs :

M. ARTHUR VIDARD, INRIA GRENOBLE - RHONE ALPES

M. DIDIER AUROUX, UNIVERSITE COTE D'AZUR

Membres du jury :

MME HÉLÈNE ROUX, TOULOUSE INP, Président

M. EHOUARN SIMON, TOULOUSE INP, Invité

MME MÉLANIE ROCHOUX, CERFACS, Membre

M. PIERRE TANDEO, IMT ATLANTIQUE, Membre

M. SERGE GRATTON, TOULOUSE INP, Membre

Table des matières

Table des matières 1

Liste des figures 4

Liste des tableaux 6

Liste des algorithmes 7

Remerciements 8

Introduction et motivation 11

I État de l'art de l'optimisation en lien avec l'assimilation de données 14

1 Assimilation de données et régularisation 15
1.1 Conventions et notations de l'assimilation de données 15
1.2 Méthodes d'assimilation de données 18
1.2.1 Point de vue statistique (Filtre de Kalman, EnKF ...) 18
1.2.2 Approche variationnelle (4DVar) 24
1.3 Régularisation et assimilation de données 32
1.3.1 Le 4DVar comme une régularisation de Tikhonov des observations 32
1.3.2 Les régularisations LASSO, Elastic Net et Fused-LASSO 33
1.3.3 La régularisation en norme L_p 35
1.3.4 Choix des paramètres de régularisation 36

2 Optimisation et assimilation de données : algorithmes de minimisation du 4DVar 41
2.1 Minimisation dans les espaces de Hilbert 42
2.1.1 Algorithmes de descente : généralités 42
2.1.2 Algorithme de descente de gradient 44
2.1.3 Algorithme de Newton 44

2.1.4 Algorithme de Gauss-Newton . . . 44
2.1.5 Algorithme du gradient conjugué linéaire pour la boucle interne . . 46
2.1.6 Algorithme de Krylov pour la boucle interne . . . 47
2.1.7 Algorithme du gradient conjugué non linéaire . . . 47
2.1.8 Algorithme BFGS . . . 48
2.2 Minimisation dans les espaces de Banach . . . 49
2.2.1 Géométrie des espaces de Banach . . . 49
2.2.2 Descente de gradient dans les espaces de Banach . . . 54
2.2.3 Algorithmes de gradient conjugué dans les espaces de Banach . . . 57
2.3 Conclusion . . . 59

II Contributions : utilisation de la norme L_p en assimilation de données et études de nouveaux algorithmes de minimisation adaptés à cette régularisation 60

3 La régularisation en norme L_p en assimilation de données 61
3.1 Pourquoi régulariser avec une norme L_p . . . 61
3.1.1 Intérêt de modélisation statistique . . . 61
3.1.2 Intérêts numériques de la norme L_p . . . 65
3.2 Illustration sur un problème d'assimilation de données : l'advection 1D . . . 65
3.2.1 Contexte expérimental . . . 65
3.2.2 Choix des paramètres λ et p . . . 69
3.2.3 Résultats pour le scénario parfait . . . 71
3.2.4 Résultats pour le modèle imparfait . . . 73
3.2.5 Bilan de l'expérience d'advection . . . 79
3.3 Conclusion . . . 83

4 Algorithmes de minimisation du 4DVar pénalisé en norme L_p 85
4.1 Motivations pour effectuer la minimisation dans un espace non euclidien . . . 86
4.2 Comparaison théorique des différentes descentes de gradient dans les espaces de Banach . . . 87
4.2.1 Panorama des algorithmes . . . 87
4.2.2 Descente de gradient dans le dual avec recherche linéaire . . . 88
4.2.3 Gradient conjugué non linéaire dans l'espace dual . . . 93
4.2.4 Gradient conjugué non linéaire avec transport de la direction dans le primal . . . 97
4.3 Comparaisons expérimentales des algorithmes . . . 100
4.3.1 Comparaison par rapport au choix de β_k . . . 100
4.3.2 Comparaison des vitesses de convergence . . . 103
4.4 Convergence des algorithmes . . . 106
4.4.1 Algorithme de descente de gradient dans l'espace dual . . . 106
4.4.2 Algorithme de gradient conjugué non linéaire dans le dual . . . 110

4.4.3 Algorithme du gradient conjugué non linéaire avec transport de la direction dans le primal . . . 111
4.5 Conclusion . . . 114

5 Vers un problème d'assimilation de données plus réaliste : les équations de Barré de Saint-Venant en deux dimensions 115
5.1 Présentation du problème . . . 116
5.1.1 Dynamique du système . . . 116
5.1.2 Intégration numérique du système . . . 116
5.1.3 Système d'assimilation (génération des observations, fenêtre d'assimilations) . . . 117
5.1.4 Validation numérique . . . 119
5.2 Minimisation du 4DVar pénalisé . . . 120
5.2.1 Choix de la base $\mathbf{\Phi}$ pour la pénalisation . . . 121
5.2.2 Choix des paramètres λ et p . . . 121
5.2.3 Vitesse de convergence des algorithmes . . . 122
5.2.4 Comportement de NLCGDS . . . 124
5.3 Conclusion . . . 127

Conclusion et perspectives 137

III Annexes 142

A Comportement de l'algorithme d'Estatico *et al.* avec une recherche linéaire du pas 143

B Méthode proximale et régularisation implicite des algorithmes duaux 145
B.1 Méthode proximale . . . 145
B.2 Régularisation implicite des méthodes duales . . . 146

C Algorithmes duaux et descente miroir 151

Bibliographie 153

Liste des figures

1.1 Principe du filtrage en assimilation de données. 21
1.2 Schéma du 4DVar. 26
1.3 Représentation graphique des normes L_p. 36
1.4 Méthode de la L-curve 38

3.1 Conditions initiales parcimonieuses 67
3.2 Conditions initiales quasi-parcimonieuses 68
3.3 L-curve pour le problème d'advection 70
3.4 État analysé pour les différentes régularisation - scénario parfait 71
3.5 RMSE et MAE des états analysés - scénario parfait 73
3.6 Évolution temporelle des conditions initiales - scénario imparfait 75
3.7 Évolution de l'état analysé sans pénalisation pour le signal creux - scénario imparfait 75
3.8 Évolution de l'état analysé sans pénalisation pour le signal quasi-creux - scénario imparfait 76
3.9 Évolution temporelle des analysés pour les différentes régularisation pour le cas creux - scénario imparfait 78
3.10 Effet des régularisations sur l'analyse - scénario imparfait 80
3.11 Évolution temporelle des analysés pour les différentes régularisation pour le cas creux - scénario imparfait 81

4.1 Effet du changement de variables $(x', y') = J_q(x, y)$ sur une quadratique . 90

5.1 Conditions initiales pour les deux fenêtres d'assimilation 128
5.2 Grille d'observation. 129
5.3 Test du gradient 129
5.4 Carte de chaleur de la RMSE en fonction de λ et p. 130
5.5 RMSE et MAE pour les états analysés des différents algorithmes. 131
5.6 État analysé pour la descente de gradient dans le dual et le gradient conjugué non linéaire dans le dual. 132
5.7 Comparaison entre GDD et NLCGDS 133

5.8 Profils des états analysés par NLCGDS et par RPCG pour la première fenêtre d'assimilation 134
5.9 Comparaison entre NLCG et RPCG 135
5.10 Évolution de la RMSE au cours du temps pour RPCG et NLCGDS 136

B.1 Solution du problème aux moindre carrés sans pénalisation. 147
B.2 Régularisation implicite au cours des itérations pour les algorithmes d'ordre 1. 148
B.3 Régularisation implicite au cours des itérations pour les algorithmes d'ordre 2. 149
B.4 Seuillage doux et seuillage par l'opérateur de dualité 150

Liste des tableaux

3.1 Paramètres de discrétisation 66
3.2 Scénario parfait : RMSE et MAE pour les cas creux et quasi-creux. Le meilleur résultat de chaque ligne est souligné. 74
3.3 Scénario imparfait : RMSE et MAE pour le cas creux. Le meilleur résultat pour chaque colonne est souligné. 77
3.4 Scénario imparfait : RMSE et MAE pour le cas quasi-creux. Le meilleur résultat pour chaque colonne est souligné. 77

4.1 Nombre d'itérations pour converger en fonction de p et λ pour $\boldsymbol{\beta}_{\mathbf{k}}^{\mathbf{HS}}$. . . . 102
4.2 Nombre d'itérations pour converger en fonction de p et λ pour $\boldsymbol{\beta}_{\mathbf{k}}^{\mathbf{FR}}$. . . . 102
4.3 Nombre d'itérations pour converger en fonction de p et λ pour $\boldsymbol{\beta}_{\mathbf{k,dual}}^{\mathbf{HS}}$. . . 102
4.4 Nombre d'itérations pour converger en fonction de p et λ pour $\boldsymbol{\beta}_{\mathbf{k,dual}}^{\mathbf{FR}}$. . . 102
4.5 Nombre d'itérations pour converger en fonction de p et λ pour $\boldsymbol{\beta}_{\mathbf{k}} = \mathbf{0}$. . . 102
4.6 Nombre d'itérations pour converger en fonction de p et λ pour les algorithmes de descente de gradient. 104
4.7 Nombre d'itérations pour converger en fonction de p et λ pour les algorithmes de type gradient conjugué non linéaire. 105

5.1 Nombre de tours de boucle requis pour la recherche de pas au cours des itérations de GDD et NLCGDS (première fenêtre d'assimilation). 125

A.1 Étude de la longueur du pas pour l'algorithme d'Estatico *et al.* sans recherche linéaire 143
A.2 Étude de la longueur du pas pour l'algorithme d'Estatico avec recherche linéaire 144

Liste des algorithmes

1 Algorithme du 4DVar incrémental 27
2 Algorithme du gradient conjugué classique 46
3 Algorithme du gradient conjugué non-linéaire 48
4 Algorithme de Schöpfer *et al.* 55
5 Algorithme du gradient conjugué d'Herzog et Wollner 57
6 Algorithme du gradient conjugué d'Estatico *et al.* 58
7 Algorithme de la descente de gradient dans le dual avec recherche de pas linéaire 93
8 Algorithme du gradient conjugué non linéaire avec itérations dans le dual et recherche du pas linéaire 96
9 Algorithme du gradient conjugué non linéaire avec transport de la direction dans l'espace primal 99

Remerciements

« *Nobody understands quantum L_p-norm regularization* », R. Feynman.

Je souhaite commencer par remercier les deux personnes qui ont rendues cette thèse possible : mes directeurs Serge Gratton et Ehouarn Simon. J'ai énormément de chance de vous avoir eu comme encadrants ; pour vos connaissances, votre pédagogie et votre bienveillance.

Je remercie tous les amis du bureau qui ont rendus les journées de travail beaucoup plus belles. Certains avaient écrits à mon insu leurs propres remerciements (possiblement légérement exagérés) que je laisse tels quels : « Je remercie mes dieux, Rémy et Bastien, sans qui ma vie n'aurait pas la même saveur. Je remercie mon ange gardien Sadok pour la guidance spirtituelle qu'il m'offerte. Je remercie finalement Théo et Antoine, source de toutes les idées pertinentes présentées dans cette thèse ». Toutes les personnes, grandes et petites, du troisième étage étaient adorables.

J'ai adoré pouvoir faire des références à longueur de journée au seigneur des anneaux avec Bastien, apprendre la cuisine camerounaise avec Boris, Firmin et Kevin, et terminer derniers aux quizs avec Réhiii. Sophie, j'aurais aimé que tu sois venue dans notre bureau beaucoup plus tôt. Un merci tout particulier à mon compère Olivier qui m'a apporté une aide précieuse tout au long de ma thèse, et à qui je laisse désormais monopole des jeux de mots au bureau. Je remercie également Bernard Bonnard pour son humanité.

Gros Merci à Valentin pour tout ce que tu as fait pour moi et à Delphine ; parfois, une soirée plaintes-burgers c'est ce qu'il y a de mieux pour le moral. Merci aussi à Fabien pour nos sparrings et nos sessions geekage, même si tu joues Protoss.

Je remercie les amis de toujours : Vincent, Pierre, Valentin, Baptiste. Tous ceux que je n'ai pas eu la chance de revoir ces trois dernières années, mais que, j'en suis sûr, je retrouverai et tous ceux que j'ai oubliés...

Enfin et surtout je remercie la meilleure maman du monde : la mienne, et bien sûr, comme j'aime à les appeler, mon frérôt et ma soeurette.

Introduction et motivation

L'assimilation de données est l'art d'exploiter, d'une part, un modèle physique, et, d'autre part, des observations afin de prédire de la manière la plus précise possible l'évolution d'un certain système. En effet, le modèle physique seul est souvent imparfait, il souffre d'approximations de la réalité : toutes les forces et toutes les interactions entre les phénomènes en jeu sont bien trop nombreuses et complexes pour être toutes prises en compte (penser aux phénomènes météorologiques). En parallèle, les observations sont en nombre souvent largement insuffisant devant la taille des paramètres à évaluer lorsque l'on étudie des procédés géophysiques ou océanographiques. Elles peuvent être obtenues par de nombreuses méthodes, directes ou indirectes : sondes, télédétection, radiomètres, imagerie infrarouge, bouées, animaux marins etc. De plus, ces observations peuvent elles-mêmes être entachées d'erreurs (par exemple dues aux incertitudes de mesure). Elles sont néanmoins d'autant plus cruciales que les systèmes étudiés dépendent fortement des conditions initiales : une légère erreur sur celles-ci peut conduire à des scénarios complètement différents même sur des échelles de temps relativement courtes. Elles servent alors à nous informer en temps réelle si la trajectoire prédite est vraisemblable ou non.

Il est donc nécessaire de combiner ces deux aspects complémentaires, tout en cherchant à estimer la part d'erreur commise par chacun. De la navigation aérospatiale à la météorologie en passant par l'économétrie, l'assimilation de données a connu des applications dans de nombreux domaines [3]. Elle est aujourd'hui essentielle pour la gestion de ressources telles que l'eau ou la production des énergies renouvelables : il est critique de connaître et d'adapter les prévisions pour pouvoir évaluer l'énergie que les fermes solaires ou les éoliennes seront en mesure de fournir. Dans un contexte de réchauffement climatique et de dépérissement des ressources on imagine alors facilement l'importance des techniques développées dans ce domaine.

Les premières méthodes d'assimilation, dites de corrections successives, ne prenaient en compte aucun aspect probabiliste : l'importance attribuée aux observations était la même que celle attribuée au modèle [1]. L'utilisation de statistiques est introduite par les méthodes d'interpolation optimum [2] et des méthodes variationnelles [19] (3DVar, 4DVar ...). Ces dernières consistent en la minimisation d'un problème aux moindres carrés qui peut être interprété sous certaines hypothèses statistiques. Le problème devient alors fon-

damentalement un problème d'optimisation numérique. Conjointement s'est développé le filtre de Kalman qui vise à minimiser la variance des erreurs commises sur la solution. Ces diverses approches ne sont pas sans lien ; l'approche variationelle et celle de Kalman fournissent effectivement la même solution sous des hypothèses d'erreurs gaussiennes. Plus tard, ce filtre fut enrichi par Evensen qui proposa le filtre de Kalman d'ensemble (EnKF) dont l'idée principale est d'utiliser un ensemble de particules pour approximer l'état et les erreurs du système. La volonté de combiner la méthode variationnelle et celle de l'EnKF s'est traduite au travers des méthodes dites hybrides qui vise à exploiter les avantages des deux approches.

Cette thèse se concentre sur l'aspect variationnel de l'assimilation de données, et plus particulièrement sur l'utilisation de la régularisation du 4DVar par une norme L_p avec $1 < p < 2$. La régularisation permet d'une manière générale de sélectionner une solution particulière pour un problème de minimisation mal posé, au sens où il existe une infinité de solutions. Ce cas est fréquent en assimilation de données, notamment à cause du faible nombre d'observations disponibles relativement à la dimension de l'espace du vecteur d'état. Il s'agit en l'occurrence d'une manière d'injecter de l'information au problème. Initialement, le terme d'écart à l'ébauche du 4DVar était déjà un terme de régularisation vis-à-vis du terme d'écart aux observations. Un autre terme de pénalisation sur la variable d'état peut être ajouté : [33] propose ainsi d'utiliser une norme L_1 (autrement appelé régularisation LASSO). Cette norme pénalise particulièrement les composantes d'amplitude faible et est donc adaptée à la reconstruction de signaux parcimonieux (avec des nombreuses composantes nulles). Le cas $p = 2$ correspond quant à lui à la régularisation bien connue de Tikhonov (ou « ridge regression ») qui est plus adaptée pour les signaux lisses. Prendre $1 < p < 2$ apparaît alors comme un compromis entre ces extrêmes et semble convenir à des signaux qu'on peut désigner de « quasi-creux ». De tels signaux, dont la parcimonie n'est pas assez forte pour justifier d'une pénalisation en norme L_1 mais suffisamment prononcée pour que la norme L_2 soit inadaptée, apparaissent effectivement dans nombre d'applications (concentration de la sargasse ou du phytoplancton dans la mer, fronts météorologiques [33]...).

En outre, le besoin de modéliser le comportement statistique des variables par d'autres lois que la loi normale (qui est certes pratique d'un point de vue calculatoire) a récemment pris de l'ampleur - comme pour le cas de la dérivée de l'épaisseur de la glace dans la mer de Beaufort. Si la régularisation en norme L_2 est liée d'un point de vue bayésien à des hypothèses gaussiennes et la norme L_1 à une loi de Laplace, nous montrerons que la norme L_p est en lien avec la loi gaussienne généralisée, que ce soit pour le terme d'écart aux observations comme pour le terme de pénalisation ajouté sur la variable d'état. Cette loi est par exemple utilisée dans la description de la propagation d'ondes acoustiques subaquatiques [83].

Le cas de la minimisation d'une fonctionnelle lisse pénalisée par une norme L_2 ou L_1 a été profondément étudié, et des algorithmes capables d'affronter la très grande dimension

qui caractérise les problèmes d'assimilation de données ont été proposés. Les algorithmes de type gradient conjugué, Lanczos ou de gradient proximal ne sont néanmoins pas capables de prendre en compte le cas d'une norme L_p avec $p \in]1;2[$ efficacement. La prochaine étape logique est donc de proposer des algorithmes qui le puissent. Or, le cadre mathématique adéquat pour ce type de régularisation est celui des espaces de Banach. Nous nous appuierons donc sur les algorithmes conçus pour la minimisation de fonctionnelles dans les espaces L_p pour, en définitive, offrir des algorithmes de type descente de gradient et gradient conjugué non linéaire adaptés à l'utilisation de la norme correspondante.

Ce manuscrit s'articule de la manière suivante : la première partie présente les diverses approches de l'assimilation de données et les méthodes de régularisation possibles, avant de se concentrer sur les algorithmes utilisés dans les centres opérationnels pour minimiser la fonctionnelle du 4DVar. On y introduit également l'optimisation dans les espaces de Banach, non exploitée actuellement par la communauté de l'assimilation. Le chapitre 3 a pour but de justifier la régularisation en norme L_p d'un point de vue statistique, puis numériquement sur l'exemple simple d'un problème d'advection en une dimension. On présente alors dans le chapitre suivant différents algorithmes pour minimiser le 4DVar pénalisé par une telle norme. Nous les comparons entre eux et étudions leurs propriétés de convergence. Finalement, la régularisation et ces algorithmes sont appliqués dans un cadre plus réaliste d'assimilation de données basé sur les équations de Barré de Saint-Venant en deux dimensions.

Première partie

État de l'art de l'optimisation en lien avec l'assimilation de données

Chapitre 1

Assimilation de données et régularisation

1.1 Conventions et notations de l'assimilation de données 15
1.2 Méthodes d'assimilation de données 18
1.2.1 Point de vue statistique (Filtre de Kalman, EnKF ...) 18
1.2.2 Approche variationnelle (4DVar) 24
1.3 Régularisation et assimilation de données 32
1.3.1 Le 4DVar comme une régularisation de Tikhonov des observations 32
1.3.2 Les régularisations LASSO, Elastic Net et Fused-LASSO 33
1.3.3 La régularisation en norme L_p 35
1.3.4 Choix des paramètres de régularisation 36

Nous commençons par rappeler le formalisme de l'assimilation de données : quel est son but, quelles sont les méthodes existantes qui permettent de l'atteindre ainsi que les interprétations statistiques sous-jacentes. Parallèlement, nous rappelons les principes de la régularisation dans un contexte général et son utilisation actuelle en assimilation de données.

1.1 Conventions et notations de l'assimilation de données

Nous avons à disposition un ensemble d'équations décrivant un phénomène physique, qui est observé à divers instants et endroits. Nous voulons combiner au mieux ces deux sources d'informations, théorique pour l'une et expérimentale pour l'autre, et peser judicieusement leur part de précision vis à vis de la réalité.

Nous définissons d'abord les différents objets mathématiques intervenant en assimilation de données. Le formalisme des notations correspond à celui introduit par [5] : les minuscules italiques représenteront des valeurs scalaires, les majuscules ($\mathcal{M}$, $\mathcal{H}$, ...) désigneront des opérateurs non linéaires. Des minuscules grasses ($\mathbf{y}$) seront utilisées pour

représenter des vecteurs tandis que les majuscules grasses ($\mathbf{M}$, $\mathbf{H}$) représenteront des matrices ou des opérateurs linéaires.

Le vecteur d'état x

C'est la variable d'intérêt : $\mathbf{x}_i$ représente l'état du système au temps t_i (hauteur, masse volumique d'un fluide etc.). Il comporte également un exposant : x^a, x^f, x^b, x^t . L'exposant a correspond l'état analysé (produit par l'assimilation), le f représente la prévision (forecast), le b l'ébauche (background, souvent utilisé comme point de départ des algorithmes) et enfin le t représentera l'état vrai (true).

Le modèle M

Il décrit l'évolution du fluide et s'écrit sous la forme d'un système continu d'équations aux dérivées partielles souvent non linéaires. De façon semi-discrétisée en espace :

$$\begin{cases} \frac{\partial \mathbf{x}(t)}{\partial t} = M(\mathbf{x}(t)) \quad \text{sur } \Omega \\ \mathbf{x}(t) = \mathbf{x}_0 \end{cases}$$

Le modèle M peut être linéarisé au voisinage de x_0 et on pose :

$$\mathbf{M} = \frac{\partial M}{\partial x}(\mathbf{x}_0).$$

Le vecteur d'observations y

Le vecteur y_i contient les observations faites au temps t_i qui proviennent en pratique de sources multiples (satellites, bouées, données in-situ...).

L'opérateur d'observation discret $\mathcal{H}$

Cet opérateur permet de relier l'espace des états à l'espace des observations. En effet, les observations ne sont pas nécessairement localisées sur l'ensemble des points de la grille de discrétisation. De plus, elles peuvent également être de nature différente des variables du modèle (par exemple, lors de la télédétection, on observe le domaine fréquentiel d'ondes émises ou réfléchies pour en déduire des propriétés d'objets à distance, comme leur composition chimique, leur densité etc.). Elles sont reliées à l'état continu $\mathbf{x}^c$ par :

$$\mathbf{y}_i = \mathcal{H}_i^c(\mathbf{x}_i^c) + \varepsilon_i^m$$

avec $\mathcal{H}_i^c$ l'observateur d'observation continu au temps t_i et ε_i^m les erreurs de mesure à ce même instant. Nous supposons que ces erreurs sont indépendantes de l'état continu. L'état vrai $\mathbf{x}^t$ correspond à la projection de l'état continu $\mathbf{x}^c$ sur l'espace du vecteur d'état :

$$\mathbf{x}_i^t = \mathbf{\Pi x_i^c}.$$

Il représente ainsi la réalité discrétisée. Nous introduisons l'opérateur d'observation discret $\mathcal{H}_i$ au temps t_i. L'observation $\mathbf{y}_i$ s'exprime alors par

$$\mathbf{y}_i = \mathcal{H}_i(\mathbf{x}_i^t) + \varepsilon_i$$

avec ε_i l'erreur d'observation. Elle est définie comme la somme de l'erreur de mesure ε_i^m et d'une erreur dite de représentativité ([4]) notée ε_i^r correspondant aux erreurs engendrées par la représentation dans un espace discret de la réalité continue. Elle s'écrit :

$$\varepsilon_i^r = \mathcal{H}_i^c(\mathbf{x}_i^c) - \mathcal{H}_i(\mathbf{x}_i^t).$$

Cet opérateur peut être linéarisé autour d'un point de la trajectoire via la formule :

$$\mathbf{H}_i = \frac{\partial \mathcal{H}_i}{\partial x}(\mathbf{x}_i).$$

La matrice de covariance d'erreur d'observation $\mathbf{R}_i$

La matrice de covariance de l'erreur d'observation $\mathbf{R}_i$ à l'instant t_i est définie par

$$\mathbf{R}_i = E[\varepsilon_i \varepsilon_i^T]$$

avec ε_i l'erreur d'observation à l'instant t_i. $\mathbf{R}_i$ est donc semi-définie positive et même définie positive en pratique (c'est à dire qu'il n'existe aucune relation affine presque sûre entre les composantes du vecteur aléatoire ε_i). Nous pouvons alors définir un produit scalaire $\langle \cdot, \cdot \rangle_{\mathbf{R}^{-1}}$ sur l'espace des observations $\mathcal{O}$ par :

$$\forall (\mathbf{u}, \mathbf{v}) \in \mathcal{O}^2, \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{R}^{-1}} = \langle \mathbf{u}, \mathbf{R^{-1}v} \rangle$$

où $\langle \cdot, \cdot \rangle$ est le produit scalaire euclidien sur $\mathcal{O}$ et $\mathbf{R}$ est la matrice de covariance d'erreur d'observation, supposée fonction du temps. De ce produit scalaire est issue la norme

$$\|\mathbf{u}\|_{\mathbf{R}^{-1}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{R}^{-1}}}.$$

La matrice de covariance d'erreur d'ébauche B

Nous appelons l'ébauche $\mathbf{x}^b$ la condition initiale de notre problème d'assimilation que l'on a a priori. C'est par exemple, pour un nouveau cycle d'assimilation, l'état $\mathbf{x}^a$ résultant de la dernière analyse. L'erreur d'ébauche se définit alors par :

$$\varepsilon^b = \mathbf{x}^b - \mathbf{x}^t$$

(où $\mathbf{x}^t$ correspond à l'état vrai). La matrice de covariance d'erreur d'ébauche est définie par

$$\mathbf{B} = E[\varepsilon^b(\varepsilon^b)^T].$$

De même que pour les observations, nous définissons le produit scalaire $\langle \cdot, \cdot \rangle_{\mathbf{B}}$ sur l'espace des états X par :

$$\forall(\mathbf{u}, \mathbf{v}) \in X^{\mathbf{2}}, \quad \langle \mathbf{u}, \mathbf{v} \rangle_{\mathbf{B}^{-1}} = \langle \mathbf{u}, \mathbf{B}^{-\mathbf{1}}\mathbf{v} \rangle$$

dont est issue la norme

$$\|\mathbf{u}\|_{\mathbf{B}^{-1}} = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle_{\mathbf{B}^{-1}}}.$$

1.2 Méthodes d'assimilation de données

Nous présentons succinctement dans cette partie les méthodes liées à l'approche statistique et variationnelle de l'assimilation de données. Bien que cette thèse s'intéresse particulièrement à l'aspect variationnel, nous mentionnons les autres techniques possibles, d'autant que ces deux approches entretiennent des liens étroits et sont même équivalentes dans certains cadres. Les méthodes stochastiques d'assimilation de données reposent essentiellement sur le filtre de Kalman [6], qui est un filtre optimal pour des problèmes linéaires. Le terme filtrage signifie que seules les observations passées et présentes sont prises en compte pour l'estimation de l'état le plus probable du système (contrairement par exemple à un « lissage » qui prend aussi en compte les observations futures). Une description détaillée de l'approche statistique de l'assimilation de données et des différentes variantes du filtrage de Kalman peut être trouvée dans par exemple [7] ou encore [11].

1.2.1 Point de vue statistique (Filtre de Kalman, EnKF ...)

Il s'agit d'ajuster l'état du modèle afin qu'il coïncide le mieux (en un sens à préciser) avec les observations. Commençons par regarder les deux estimateurs les plus courants : le BLUE et le MAP.

Recherche du BLUE (Best Linear Unbiased Estimator)

Comme son nom l'indique cet estimateur fait l'hypothèse d'opérateurs d'observations $\mathbf{H}$ et de modèle $\mathbf{M}$ linéaires. Une première estimation pour l'analyse repose de plus sur l'hypothèse d'une dépendance linéaire entre $\mathbf{x}_a$, l'ébauche $\mathbf{x}_b$ et les observations $\mathbf{y}$: on cherche des matrices $\mathbf{L}$ et $\mathbf{K}$ telles que

$$\mathbf{x}^a = \mathbf{L}\mathbf{x}^b + \mathbf{K}\mathbf{y}.$$

Ajoutons l'hypothèse que les erreurs sont non biaisées, i.e. $E[\mathbf{e}^b] = E[\mathbf{x}^b - \mathbf{x}^t] = 0$, $E[\mathbf{e}^o] = E[\mathbf{y} - \mathbf{H}\mathbf{x}^t] = 0$ et $E[\mathbf{e}^a] = E[\mathbf{x}^a - \mathbf{x}^t] = 0$. Ces conditions fournissent $\mathbf{L} = \mathbf{I} - \mathbf{K}\mathbf{H}$ puis $\mathbf{e}^a = \mathbf{e}^b + \mathbf{K}(\mathbf{e}^o - \mathbf{H}\mathbf{e}^b)$. Nous pouvons maintenant exprimer la matrice de covariance de l'erreur d'analyse sous la forme :

$$\mathbf{P}^a = (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{B}(\mathbf{I} - \mathbf{K}\mathbf{H})^T + \mathbf{K}\mathbf{R}\mathbf{K}^T.$$

Le meilleur estimateur, au sens où il minimise la trace de $\mathbf{P}^a$, est alors donné par

$$\mathbf{x}^a = \mathbf{x}^b + \mathbf{K}(\mathbf{y} - \mathbf{H}\mathbf{x}^b) \tag{1.1}$$

avec la matrice dite de gain

$$\mathbf{K} = \mathbf{B}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)^{-1}.$$

Recherche du maximum a posteriori

D'une manière générale, l'estimateur du maximum a posteriori (MAP) pour le paramètre $\boldsymbol{\theta}$, étant donné un vecteur $\mathbf{d}$, est le paramètre qui maximise la log-vraisemblance $L(\boldsymbol{\theta}) = \log p_X(\mathbf{d}; \boldsymbol{\theta})$. Lorsque $\mathbf{d}$ est la réalisation d'un vecteur gaussien $X \sim \mathcal{N}(\mathbf{A}\boldsymbol{\theta}, \boldsymbol{C})$, la log-vraisemblance est $-\frac{1}{2}(d - \mathbf{A}\boldsymbol{\theta})^T\mathbf{C}^{-1}(\mathbf{d} - \mathbf{A}\boldsymbol{\theta}) + \log(\text{Constante})$ et $\boldsymbol{\theta}$ est solution du problème aux moindres carrés $\min \|\mathbf{A}\boldsymbol{\theta} - \mathbf{d}\|_{\mathbf{C}^{-1}}$. De même, si $\mathbf{A}$ est non linéaire, $\boldsymbol{\theta}$ est solution du problème aux moindres carrés $\min \|A(\boldsymbol{\theta}) - \mathbf{d}\|_{\mathbf{C}^{-1}}$.

Dans notre cas l'estimateur du maximum a posteriori va maximiser $p_{Y|X}(\mathbf{y}|\mathbf{x}) = \frac{p_{X,Y}(\mathbf{x},\mathbf{y})}{p_X(\mathbf{x})}$ étant données les observations contenues dans $\mathbf{y}$ (loi de Bayes). Considérons le vecteur aléatoire $\boldsymbol{Y} = \mathbf{H}\boldsymbol{X} + \boldsymbol{N}$ avec $\boldsymbol{X}$ suivant une loi gaussienne centrée sur l'ébauche : $\boldsymbol{X} \sim \mathcal{N}(\mathbf{x}^b, \mathbf{B})$ et $\boldsymbol{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. Notons $\mathbf{y}$ une réalisation de $\boldsymbol{Y}$. Alors le MAP minimise

$$\frac{1}{2}(\mathbf{x} - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x} - \mathbf{x}^b) + \frac{1}{2}(\mathbf{y} - \mathbf{H}\mathbf{x})^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x})$$

et est donné par

$$\mathbf{x}^{MAP} = \mathbf{x}^b + (\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} + \mathbf{B}^{-1})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b) \tag{1.2}$$

La formule matricielle de Sherman-Morrison

$$(\boldsymbol{A} + \boldsymbol{U}\boldsymbol{V}^T)^{-1} = \boldsymbol{A}^{-1} - \boldsymbol{A}^{-1}\boldsymbol{U}(\boldsymbol{I} + \boldsymbol{V}^T\boldsymbol{A}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^T\boldsymbol{A}^{-1}$$

permet d'écrire

$$\mathbf{x}^{MAP} = \mathbf{x}^b + \mathbf{B}\mathbf{H}^T(\mathbf{R} + \mathbf{H}\mathbf{B}\mathbf{H}^T)^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b)$$

et montre, par comparaison avec (1.1), que le BLUE et le MAP sont équivalents sous des hypothèses gaussiennes. Le filtre de Kalman est un algorithme mettant à profit les estimateurs précédents pour assimiler au cours du temps de nouvelles observations et produire le meilleur estimateur correspondant.

Le filtre de Kalman

L'assimilation se déroule en deux phases : une phase d'analyse et une phase de prévision. Cette seconde étape permet de fournir une prévision de l'état courant du système. Elle va nécessiter l'emploi du modèle : l'état courant va être obtenu après intégration des équations du modèle depuis l'état analysé précédent du système (équations 1.3). Cette phase va fournir également la matrice de covariance des erreurs de prévisions, $\mathbf{P}^f$, correspondant à la propagation de l'erreur d'analyse précédente via le modèle, à laquelle on ajoute une estimation $\mathbf{Q}$ de la matrice de covariance de l'erreur de modèle.

L'étape d'analyse (équations (1.4)) va réaliser une estimation de l'état du système en corrigeant l'état courant à partir des écarts aux observations, et va fournir la matrice de covariance des erreurs d'analyse $\mathbf{P}^a$. Cela va nécessiter la connaissance de la matrice de covariance de l'erreur d'observation $\mathbf{R}$ regroupant les erreurs de mesure et de représentativité par la grille spatio-temporelle du modèle. Le modèle permettant de passer de l'état à l'instant t_k à l'état à l'instant t_{k+1}, de même que l'opérateur d'observations à l'instant t_{k+1} sont, dans ce cas de figure, supposés linéaires et sont notés respectivement $\mathbf{M}_{k,k+1}$ et $\mathbf{H}_{k+1}$. En somme l'algorithme du filtre de Kalman s'écrit :

Prévision

$$\begin{aligned} \mathbf{x}^f_{k+1} &= \mathbf{M}_{k,k+1}\mathbf{x}^a_k \\ \mathbf{P}^f_{k+1} &= \mathbf{M}_{k,k+1}\mathbf{P}^a_k\mathbf{M}^T_{k,k+1} + \mathbf{Q}_k \end{aligned} \tag{1.3}$$

Analyse

$$\begin{aligned} \mathbf{K}_{k+1} &= \mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1}(\mathbf{H}_{k+1}\mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1} + \mathbf{R}_{k+1})^{-1} \\ \mathbf{x}^a_{k+1} &= \mathbf{x}^f_{k+1} + \mathbf{K}_{k+1}(y_{k+1} - \mathbf{H}_{k+1}\mathbf{x}^f_{k+1}) \\ \mathbf{P}^a_{k+1} &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}^f_{k+1} \end{aligned} \tag{1.4}$$

La matrice $\mathbf{K}$ est appelée la matrice de gain de l'analyse statistique. C'est la matrice de gain optimale qui minimise la variance de l'erreur d'analyse. Le principe de l'assimilation via un filtre de Kalman est représenté sur la Figure 1.1. Nous avons jusqu'à présent considéré des opérateurs linéaires, mais le filtre de Kalman s'étend naturellement à des opérateurs non-linéaires en considérant successivement leur approximations linéaires (leur linéaire tangent dans le langage de l'assimilation).

Le filtre de Kalman étendu

Pour des opérateurs H et M non linéaires, on définit le linéaire tangent des opérateurs $\mathbf{H}$ et $\mathbf{M}$ par ([12]) :

$$\begin{aligned} \mathbf{H}_{k+1} &= \frac{\partial H_{k+1}}{\partial x}(\mathbf{x}^f_{k+1}) \\ \mathbf{M}_{k,k+1} &= \frac{\partial M_{k,k+1}}{\partial x}(\mathbf{x}^a_k). \end{aligned}$$

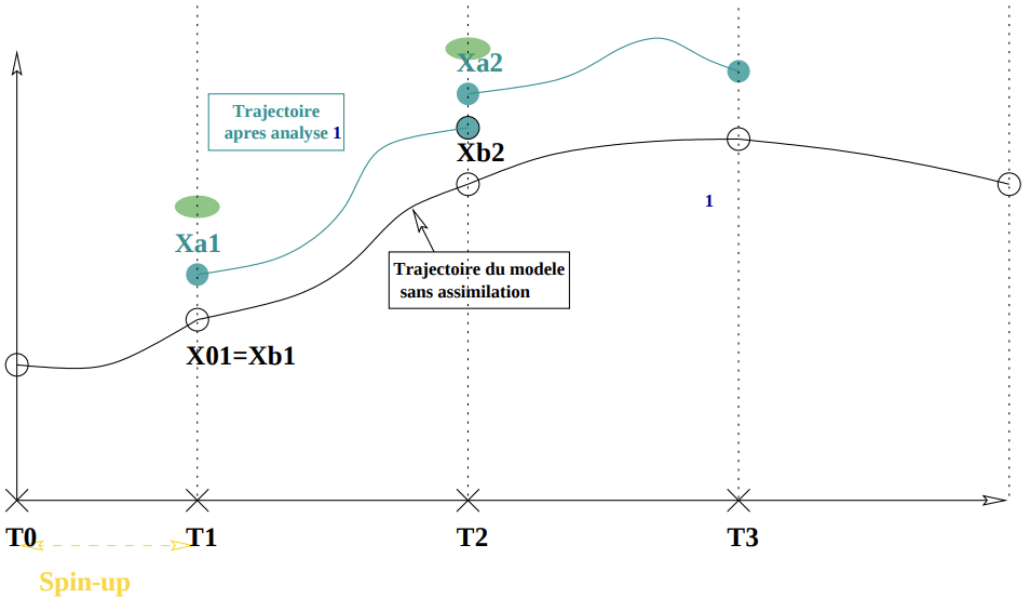


FIGURE 1.1 – Principe du filtrage en assimilation de données. Crédit pour la figure : C. Robert.

Le vecteur d'état analysé $\mathbf{x}^a$ sera alors propagé par le modèle non linéaire, tandis que les matrices de covariance d'erreur de prévision $\mathbf{P}^f$ et d'analyse $\mathbf{P}^a$ le seront par les différents modèles linéaires tangents. On obtient alors le système suivant :

Prévision

$$\begin{aligned}\mathbf{x}^f_{k+1} &= M_{k,k+1}(\mathbf{x}^a_k)\\ \mathbf{P}^f_{k+1} &= \mathbf{M}_{k,k+1}\mathbf{P}^a_k\mathbf{M}^T_{k,k+1} + \mathbf{Q}_k\end{aligned}$$

Analyse

$$\begin{aligned}\mathbf{K}_{k+1} &= \mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1}(\mathbf{H}_{k+1}\mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1} + \mathbf{R}_{k+1})^{-1}\\ x^a_{k+1} &= x^f_{k+1} + \mathbf{K}_{k+1}(y_{k+1} - H_{k+1}(x^f_{k+1}))\\ \mathbf{P}^a_{k+1} &= (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}^f_{k+1}\end{aligned}$$

On n'obtient plus alors la solution optimale (au sens où la variance de l'erreur d'analyse ne sera plus minimale) mais une solution approchée. Outre les problèmes liés à la non optimalité du filtre de Kalman étendu et à la méconnaissance des matrices $\mathbf{P}_0$, $\mathbf{Q}$ et $\mathbf{R}$ qui peuvent entraîner une certaine inefficacité du filtre, le problème lié à la taille du vecteur d'état pour des modèles réalistes (de l'ordre de $10^6 - 10^7$) rend impossible

son implémentation complète. L'idée est alors d'approcher l'espace complet des covariances d'erreur par un sous-espace de dimension réduite. Il est en effet fréquent que la quasi-totalité de la dynamique du modèle puisse être déterminée, à chaque instant, par la donnée d'un nombre limité de variables, ou de combinaisons linéaires de variables (appelés modes dominants).

Filtres de Kalman de rang réduit

Nous cherchons à effectuer les corrections seulement dans les directions (modes) où l'erreur est amplifiée par le système. Notons p le nombre de modes retenus. les différents filtres de rang réduits vont dépendre essentiellement de la façon dont on approche ces modes. De plus, les matrices de covariance d'erreur étant symétriques définies positives, elles peuvent s'écrire $\mathbf{P}_k^f = \mathbf{S}_k^f \mathbf{S}_k^{fT}$, $\mathbf{P}_k^a = \mathbf{S}_k^a \mathbf{S}_k^{aT}$ et $\mathbf{Q}_k = \mathbf{\Sigma}_k \mathbf{\Sigma}_k^T$. Ces méthodes de rang réduit ne vont plus faire intervenir directement ces matrices, mais leur racine carrée. Parmi ces filtres citons le filtre RRSQRT issu de [13], où les modes vont correspondre aux p vecteurs propres associés aux p plus grandes valeurs propres de la matrice $\mathbf{S}_k^a \mathbf{S}_k^{aT}$, ou encore le filtre SEEK proposé par Pham *et al.* [14].

Filtre de Kalman d'ensemble

Proposé par Evensen ([8, 9]), c'est également un filtre de Kalman de rang réduit. Il se base sur un ensemble de vecteurs d'états appelés particules, dont le nombre est très inférieur à la dimension du problème, mais censées échantillonner correctement la distribution des états possibles du système.

Ainsi, étant données N particules, chacune d'entre elles sera mise à jour de la même manière que l'étape (1.1) : $\forall i \in \mathbb{N}_N$

$$\mathbf{x}_i^a = \mathbf{x}_i^f + \mathbf{K}(\mathbf{y}_i - H(\mathbf{x}_i^f)).$$

où la matrice de gain $\mathbf{K} = \mathbf{P}^f \mathbf{H}^T (\mathbf{H}\mathbf{P}^f \mathbf{H}^T + \mathbf{R})^{-1}$ est ici calculée plus facilement en estimant $\mathbf{P}^f$ par

$$\mathbf{P}^f = \frac{1}{N-1} \sum_{i=1}^{N} (\mathbf{x}_i^f - \bar{\mathbf{x}}^f)(\mathbf{x}_i^f - \bar{\mathbf{x}}^f)^T$$

avec $\bar{\mathbf{x}}^f = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i^f$ l'estimateur de la moyenne des états prédits pour chaque particule. La méthode revient au même que perturber l'ébauche $\mathbf{x}^b$ N fois par un bruit gaussien. De la sorte, les erreurs accordées aux observations sont moins importantes ce qui peut rendre le filtre divergent. Nous devons de même perturber les observations par un bruit gaussien ε_i^o de moyenne nulle et de covariance $\mathbf{R}_{k+1}$: $\forall i \in \mathbb{N}_N$,

$$\mathbf{y}_{k+1}^j = \mathbf{y}_{k+1} + \varepsilon_i^o \tag{1.5}$$

et la matrice de covariance des erreurs d'observations $\mathbf{R}$ sera alors estimée par $\mathbf{R} = \frac{1}{N-1} \sum_{i=1}^{N} \varepsilon_i^o \varepsilon_i^{oT}$. Ce filtre permet de prendre en compte implicitement les non-linéarités

du modèle. Son principal défaut réside dans le nombre N relativement grand de simulations à réaliser. Néanmoins, ce filtre peut être appliqué en pratique pour des modèles réalistes, ce qui n'est pas le cas du filtre de Kalman classique, du fait de la taille des matrices de covariance d'erreur.

L'étape de perturbations des observations introduit cependant des erreurs d'échantillonnage et rend le filtre moins performant, particulièrement pour des ensembles de petites tailles. Le filtre de Kalman d'ensemble déterministique (DEnKF) développé dans [10] remédie à ce problème en ne perturbant plus les observations, mais en divisant par deux le gain de Kalman. Cette étape simple permet en pratique de ne pas réduire prématurément la dispersion de l'ensemble de particules et de retrouver asymptotiquement la valeur optimale de la matrice de covariance d'erreur de l'analyse donnée par le filtre de Kalman.

Filtre d'ensemble de rang réduit : EnSRF

Pour Ensemble Square Root Filters ([15]), la motivation pour ce filtre est entre autre d'éviter l'erreur d'échantillonnage causées par la perturbation des observations (1.5) [18]. Pour ce faire la matrice de gain $\mathbf{K}$ est substituée par une matrice

$$\tilde{\mathbf{K}} = \mathbf{P}^b\mathbf{H}^T \left((\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R})^{-\frac{1}{2}}\right)^T \left((\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R})^{\frac{1}{2}} + \mathbf{R}^{\frac{1}{2}}\right)^{-1},$$

faisant appel à une racine carrée de la matrice $\mathbf{R}$ et de la matrice $\mathbf{H}\mathbf{P}^b\mathbf{H}^T + \mathbf{R}$, et permettant de retrouver une expression pour $\mathbf{P}^a$ similaire à la mise à jour (1.4).

Inflation et localisation

Enfin, ces méthodes peuvent souffrir d'un problème de sous-échantillonnage. Mentionnons les méthodes de localisation et d'inflation qui ont été développées pour y remédier.

La première exploite le fait que pour un système synoptique, les états de deux points éloignés seront peu corrélés. Pour mettre à jour une variable, seules les observations locales seront alors utilisées. De même, en remarquant que la matrice $\mathbf{P}^f$ est de rang faible, il est possible de la modifier en faisant son produit composante par composante par une matrice de corrélation pour éliminer de fausses relations entre variables distantes (voir [16]).

La seconde, donnée dans [17], consiste en la multiplication de l'ensemble d'approximation de la matrice de covariance de l'erreur de forecast par un facteur $\gamma > 1$. Le gain de Kalman étant corrélé à l'ensemble des particules, cette technique permet d'élargir la distribution *a priori* artificiellement en empêchant les variables de devenir trop dépendantes les unes des autres (se pose alors la question d'un choix judicieux de γ).

1.2.2 Approche variationnelle (4DVar)

Introduites par Sasaki dès 1955 ([19]), les méthodes variationnelles sont basées sur la minimisation d'une fonction coût Ω mesurant les écarts entre l'état estimé et les données disponibles. Alors que dans le cadre du filtrage stochastique les observations n'étaient utilisées qu'une seule fois et n'influaient pas sur les calculs des divers estimés qui leurs étaient antérieures, l'approche variationnelle va opérer globalement sur l'ensemble des observations disponibles dans la fenêtre d'assimilation pour réaliser la minimisation. Cette approche permet ainsi de calculer la trajectoire optimale du système et non plus la meilleure estimation de l'état à un instant d'observation.

4DVar

Nous avons à dispositions toujours une ébauche de la condition initiale $\mathbf{x}^b$ ainsi que N observations faites aux instants t_i pour $1 \leq i \leq N$ (si $N = 1$, i.e. la variable temporelle n'entre pas en jeu, on parle du 3DVar). La fenêtre temporelle $[t_1; t_N]$ est appelée fenêtre d'assimilation. Nous allons chercher la condition initiale $\mathbf{x}_0$ telle que, soumise au modèle, la trajectoire obtenue soit optimale (dans un sens de moindres carrés pondérés que nous allons préciser) vis-à-vis des observations et de l'information *a priori* sur la condition initiale. Celle-ci calculée, nous obtenons l'état analysé $\mathbf{x}^a$. Une prochaine fenêtre d'assimilation commençant à t_N pourra utiliser l'intégration de $\mathbf{x}^a$ jusqu'à cet instant comme ébauche pour une nouvelle analyse. L'état analysé est obtenu en minimisant la fonction coût dite du 4DVar (problème variationnel en 3 dimensions spatiales et avec la dimension temporelle) :

$$\Omega(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \sum_{i=0}^{N} \frac{1}{2}\|\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i\|^2_{\mathbf{R}_i^{-1}} \tag{1.6}$$

$$= \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=0}^{N}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i)^T\mathbf{R}_i^{-1}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0) - \mathbf{y}_i). \tag{1.7}$$

$\mathbf{B}$ est la matrice de covariance d'erreur de l'ébauche, elle correspond à la matrice $\mathbf{P}_0^f$ du filtre de Kalman. Les matrices $\mathbf{R}_i$ correspondent toujours aux matrices de covariance d'erreur des observations aux temps t_i. En écrivant

$$\mathbf{y} = (\mathbf{y}_1, ..., \mathbf{y}_k)^T, \tag{1.8}$$

$$\hat{\mathcal{H}}(\mathbf{x}) = [\mathcal{H}_1(\mathcal{M}_{0,1}(\mathbf{x})), ..., \mathcal{H}_k(\mathcal{M}_{0,k}(\mathbf{x}))]^T, \tag{1.9}$$

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{R}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{R}_k \end{pmatrix}, \tag{1.10}$$

on peut écrire le système (1.6) de manière plus condensée :

$$\Omega(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \hat{\mathcal{H}}(\mathbf{x}_0)\|^2_{\mathbf{R}^{-1}} \tag{1.11}$$

$$\tag{1.12}$$

$\mathbf{x}^a$ peut donc être obtenu par résolution de

$$\nabla\Omega(\mathbf{x}^a(t_0)) = 0. \tag{1.13}$$

Si les opérateurs $\mathcal{H}_i$ et $\mathcal{M}$ sont linéaires, Ω est alors une quadratique strictement convexe ($\mathbf{B}$ est supposée définie positive), d'où l'unicité de la solution. Dans le cas contraire, il est d'usage de ne rechercher qu'un point critique pour des raisons de coût et de temps de calcul. De plus si le modèle est parfait, il existe une certaine équivalence entre les solutions trouvées par le 4D-Var et le filtre de Kalman : il est possible de montrer qu'en partant des mêmes données, l'analyse du 4DVar à la fin de la période d'assimilation est égale à celle du filtre de Kalman au même instant. En effet, pour des opérateurs linéaires, (1.13) mène à $\nabla\Omega(\mathbf{x}_0) = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) - \mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x})$, puis à

$$\mathbf{x}_a = \mathbf{x}^b + (\mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H})^{-1}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\mathbf{x}^b)$$

qui est la même expression que (1.2). La dérivée seconde de Ω est également égale à l'inverse de la matrice de covariance d'erreur de l'analyse du filtre de Kalman : $\nabla^2\Omega(\mathbf{x}_0) = \mathbf{B}^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = (\mathbf{P}^a)^{-1}$. Autrement dit, la précision de l'analyse du 4DVar est proportionnel à la courbure de Ω. Dans le cadre de l'océanographie ou de la météorologie, ces opérateurs ne sont pas linéaires. On utilisera alors leurs linéaires tangents dans une boucle itérative de type Gauss-Newton : c'est le principe du 4DVar incrémental. L'assimilation par le 4D-Var est schématisée figure (1.2).

4DVar incrémental

Le problème de la non linéarité des différents opérateurs entraîne un surcoût de calcul de la méthode ainsi que l'apparition de nombreux extrema locaux pouvant altérer les performances du minimiseur. Une alternative consiste à linéariser la fonction coût afin de la rendre quadratique : c'est la version incrémentale du 4D-Var proposé par Courtier *et al.* [20]. La fonction coût ne va plus être minimisée par rapport à l'état $\mathbf{x}_0$, mais par rapport à un incrément $\delta\mathbf{x}_0$ défini par $\mathbf{x}_0 = \mathbf{x}^b + \delta\mathbf{x}_0$. Les opérateurs $\mathcal{H}_i$ et $\mathcal{M}$ sont de plus linéarisés au voisinage $\mathbf{x}^b$ pour tout i :

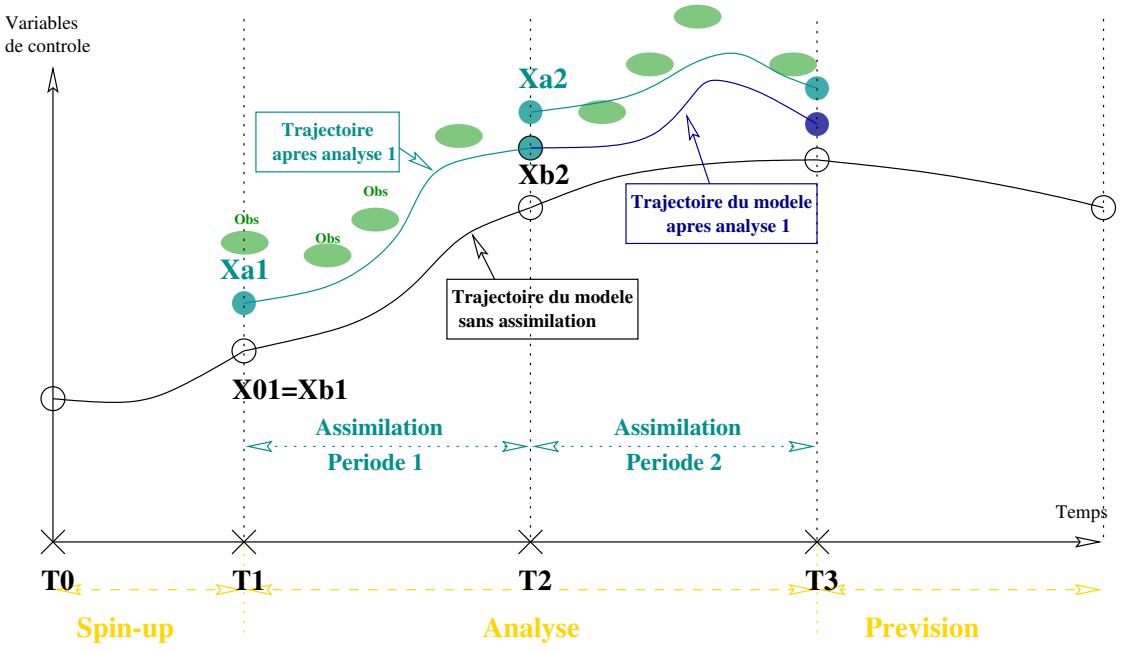


FIGURE 1.2 – Assimilation par le 4DVar. Crédit pour la figure : C. Robert.

$$\mathcal{M}_{0,i}(\mathbf{x}^b + \delta\mathbf{x}_0) \approx \mathcal{M}_{0,i}(\mathbf{x}^b) + \mathbf{M}_{0,i}\delta\mathbf{x}_0$$
$$\mathcal{H}_i(\mathbf{x}^b + \delta\mathbf{x}_0) \approx \mathcal{H}_i(\mathbf{x}^b) + \mathbf{H}_i\delta\mathbf{x}_0.$$

La nouvelle fonction coût s'écrit

$$\begin{aligned}\Omega(\delta\mathbf{x}_0) &= \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{H}_i\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i)^T\mathbf{R}_i^{-1}(\mathbf{H}_i\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i) \\ &= \frac{1}{2}\delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \frac{1}{2}(\hat{\mathbf{H}}\delta\mathbf{x}_0 - \mathbf{d})^T\mathbf{R}^{-1}(\hat{\mathbf{H}}\delta\mathbf{x}_0 - \mathbf{d}), \qquad (1.14)\end{aligned}$$

avec $\mathbf{d}_i = \mathbf{y}_i - \mathcal{H}_i(\mathcal{M}_{0,i}(\mathbf{x}^b))$ le vecteur d'innovation et

$$\hat{\mathbf{H}} = [\mathbf{H}_1(\mathbf{M}_{0,1}(\mathbf{x})), ..., \mathbf{H}_k(\mathbf{M}_{0,k}(\mathbf{x}))]^T$$

la linéarisation de la matrice augmentée $\hat{\mathcal{H}}$, définie à l'équation (1.9), autour de $\mathbf{x}_b$.

La minimisation de la fonction coût via le calcul de l'incrément optimal en linéarisant les opérateurs $\mathcal{M}$ et $\mathcal{H}$ correspond à la l'exécution d'une « boucle interne ». En pratique, cette minimisation ne sera pas menée jusqu'à l'obtention de l'optimum de (1.14) : seul un certain nombre prédéfini d'itérations sera réalisé. Afin de prendre en compte les

non-linéarités du modèle et des opérateurs d'observations, une boucle dite "externe" est ajoutée au processus. Cette dernière met à jour les vecteurs d'innovation en utilisant la dynamique complète non linéaire issue du point calculé par la boucle interne. Ceci permet de s'approcher de la minimisation de (1.6). L'algorithme correspond est :

Algorithme 1 Algorithme du 4DVar incrémental

1: **Initialisation :** $\mathbf{x}_0^0 = \mathbf{x}^b$
2: **tant que** $k \leq k_{max}$ **ou** $\|\delta \mathbf{x}^{a,k}\| < \varepsilon$ **faire (boucle externe)**
3: $\quad \mathbf{d}_i^k = \mathbf{y}_i^{obs} - \mathbf{H}_i(\mathbf{M}_{0,i}(\mathbf{x}_0^k))$
4: $\quad$ **Chercher l'incrément d'analyse** $\delta \mathbf{x}^{a,k}$ **minimisant (boucle interne)**

$$\Omega(\delta \mathbf{x}_k) = \frac{1}{2}(\delta \mathbf{x}^k)^T \mathbf{B}^{-1} \delta \mathbf{x}^k + \frac{1}{2}\sum_{i=0}^{N}(\mathbf{H}_i \mathbf{M}_{0,i} \delta \mathbf{x}^k - \mathbf{d}_i^k)^T \mathbf{R}_i^{-1}(\mathbf{H}_i \mathbf{M}_{0,i} \delta \mathbf{x}^k - \mathbf{d}_i^k)$$

5: $\quad \mathbf{x}_0^{k+1} = \mathbf{x}_0^k + \delta \mathbf{x}^{a,k}$
6: **fin tant que**

Nous verrons à la section (2.1.4) que cet algorithme est en fait équivalent à une méthode de Gauss-Newton. Précisons maintenant comment le gradient est calculé par la méthode de l'adjoint.

Méthode adjointe

Considérons un modèle d'évolution semi-discrétisé d'un système quelconque (par exemple un fluide soumis aux équations de Navier-Stokes) :

$$\begin{cases} \dfrac{\partial \mathbf{x}(t)}{\partial t} = \mathcal{M}(\mathbf{x}(t)) \quad \text{sur } \Omega \\ \mathbf{x}(0) = \mathbf{x}_0 \end{cases}$$

avec $\mathbf{x}$ le vecteur d'état du système. La variable de contrôle dans le cas présent sera la condition initiale $\mathbf{x}_0$. Par soucis de clarté, seule cette variable de contrôle est choisie ici ; il est cependant possible d'en contrôler d'autres, par exemple certains paramètres mal connus.

La fonction coût Ω s'exprime alors comme la somme d'un terme de régularisation Ω_b mesurant l'écart à l'ébauche et d'un terme mesurant l'écart aux observations Ω_O :

$$\Omega(\mathbf{x}_0) = \frac{1}{2}\int_0^T \|\mathbf{H}\mathbf{x}(t) - \mathbf{y}(t)\|^2_{\mathbf{R}^{-1}} dt + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} \tag{1.15}$$

$$= \Omega_O(\mathbf{x}_0) + \Omega_b(\mathbf{x}_0). \tag{1.16}$$

Rechercher le minimum de cette fonction va nécessiter l'étude de ses points critiques, i.e. rechercher les points $\mathbf{x}_0^*$ vérifiant :

$$\nabla\Omega(\mathbf{x}_0^*) = 0 = \nabla\Omega_b(\mathbf{x}_0^*) + \nabla\Omega_0(\mathbf{x}_0^*).$$

Le gradient du terme Ω_b étant simple à calculer, intéressons-nous à celui du terme Ω_O. La fonction Ω_O ne dépendant pas explicitement de la variable de contrôle $\mathbf{x}_0$, le calcul direct du gradient s'avère ardu voire impossible. Cependant, ce calcul est réalisable via la méthode de l'adjoint proposée par Lions en 1968 [23] et appliquée pour la première fois en météorologie par Le Dimet en 1982 [24], puis par Le Dimet et Talagrand en 1986 [25]. En voici une description sommaire.

Considérons $\hat{\mathbf{x}}$ la dérivée de Gâteaux de $\mathbf{x}$ dans la direction h, et $\hat{\Omega}$ celle de Ω. Il vient alors

$$\begin{cases} \dfrac{\partial\hat{\mathbf{x}}(t)}{\partial t} = \dfrac{\partial\mathcal{M}}{\partial\mathbf{x}}(\mathbf{x}(t)) \cdot \hat{\mathbf{x}}(t) \quad \text{sur } \Omega \\ \hat{\mathbf{x}}(0) = h \end{cases}$$

$$\hat{\Omega}(\mathbf{x}_0, h) = \int_0^T \langle \mathbf{H}\mathbf{x}(t) - \mathbf{y}(t), \mathbf{H}\hat{\mathbf{x}}(t)\rangle dt. \tag{1.17}$$

Introduisons $\mathbf{p}(t)$ la variable adjointe. En effectuant le produit scalaire canonique entre l'équation précédente et $\mathbf{p}(t)$ puis en intégrant sur $[0;T]$ il vient :

$$\int_0^T \langle \frac{\partial\hat{\mathbf{x}}(t)}{\partial t}, \mathbf{p}(t)\rangle dt = \int_0^T \langle \frac{\partial M}{\partial\mathbf{x}}(\mathbf{x}(t)) \cdot \hat{\mathbf{x}}(t), \mathbf{p}(t)\rangle dt.$$

Après une intégration par parties, il vient :

$$\langle \hat{\mathbf{x}}(T), \mathbf{p}(T)\rangle - \langle \hat{\mathbf{x}}(0), \mathbf{p}(0)\rangle = \int_0^T \langle \hat{\mathbf{x}}(t), \left[\frac{\partial\mathcal{M}}{\partial\mathbf{x}}(\mathbf{x}(t))\right]^T \mathbf{p}(t) + \frac{\partial\mathbf{p}}{\partial t}(t)\rangle dt. \tag{1.18}$$

Si l'on définit $\mathbf{p}$ comme la solution du modèle adjoint suivant :

$$\begin{cases} \dfrac{\partial\mathbf{p}(t)}{\partial t} + \left[\dfrac{\partial\mathcal{M}}{\partial x}(\mathbf{x}(t))\right]^T \mathbf{p}(t) = \mathbf{H}^T(\mathbf{H}\mathbf{x}(t) - \mathbf{y}(t)) \\ \mathbf{p}(T) = 0 \end{cases} \tag{1.19}$$

l'équation (1.18) devient alors :

$$-\langle \hat{\mathbf{x}}(0), \mathbf{p}(0)\rangle = \int_0^T \langle \mathbf{H}\mathbf{x}(t) - \mathbf{y}(t), \mathbf{H}\hat{\mathbf{x}}(t)\rangle dt.$$

Il ne reste plus qu'à identifier avec l'équation (1.17) de $\hat{\Omega}(\mathbf{x}_0, h)$, sachant que $\hat{\Omega}(\mathbf{x}_0, h) = \langle \nabla\hat{\Omega}(\mathbf{x}_0), h\rangle$ pour obtenir :

$$\nabla\Omega(\mathbf{x}_0) = -\mathbf{p}(0).$$

En intégrant de manière rétrograde le modèle adjoint (1.19), il est donc possible d'avoir accès au gradient de la fonction coût. **On peut donc minimiser Ω par une méthode de gradient**.

Le système d'optimalité s'écrit :

$$\begin{cases} \dfrac{\partial \mathbf{x}(t)}{\partial t} = \mathcal{M}(\mathbf{x}(t)) \\ \mathbf{x}(0) = \mathbf{x}_0 \\ \dfrac{\partial \mathbf{p}(t)}{\partial t} + [\dfrac{\partial \mathcal{M}}{\partial \mathbf{x}}(\mathbf{x}(t))]^T \mathbf{p}(t) = \mathbf{H}^T(\mathbf{H}\mathbf{x}(t) - \mathbf{y}(t)) \\ \mathbf{p}(T) = 0 \\ \nabla\Omega(\mathbf{x}_0) = -\mathbf{p}(0) = 0 \end{cases}$$

Il faut noter que cette méthode permet d'obtenir le gradient de Ω de façon exacte, et ce pour des coûts de calcul assez faibles en grande dimension par rapport à une méthode de différences finies : seules deux simulations (une pour le modèle direct et une pour le modèle adjoint) sont nécessaires. L'autre approche « simpliste », où l'on évaluerait le gradient par des taux d'accroissement $\frac{\partial\Omega}{\partial \mathbf{x}_i}(\mathbf{x}_0) \approx \frac{\Omega(\mathbf{x}_0+\alpha\mathbf{e}_i)-\Omega(\mathbf{x}_0)}{\alpha}$ nécessiterait un nombre de simulations du modèle direct de l'ordre de la dimension de l'espace de contrôle et ne fournirait pas un résultat exact.

En pratique l'adjoint comme le linéaire tangent peuvent être obtenus à partir de la différentiation automatique du code calculant le modèle, par exemple grâce au logiciel TAPENADE développé par l'INRIA [26]. Il est possible de dériver l'adjoint du modèle discrétisé ou de trouver l'adjoint du modèle continu avant de discrétiser, ces deux approches étant en général aussi efficaces l'une que l'autre (une comparaison ayant été faite par A.S. Lawless *et. al.* [27]).

Le rôle de la matrice de covariance d'erreur d'ébauche B

La minimisation de la fonctionnelle de la boucle interne du $4DVar$ incrémental amène à

$$\delta\mathbf{x}_0^a = \left[\mathbf{B}^{-1} + \mathbf{M}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{M}\right]^{-1}\mathbf{M}^T\mathbf{H}^T\mathbf{R}^{-1}\mathbf{d}$$

en notant $\mathbf{d}$, $\mathbf{M}$, $\mathbf{H}$ et $\mathbf{R}$ sous forme augmentée (cf. (1.10)). L'incrément peut aussi s'écrire en utilisant la formule de Sherman-Morrison :

$$\delta\mathbf{x}_0^a = \mathbf{B}\mathbf{M}^T\mathbf{H}^T\left[\mathbf{R} + \mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^T\mathbf{H}^T\right]^{-1}\mathbf{d}.$$

La correction apportée à $\mathbf{x}_0$ est donc l'image par $\mathbf{B}$ d'une somme de différentes innovations pondérées appartenant à l'espaces des observations, $\left[\mathbf{R}+\mathbf{H}\mathbf{M}\mathbf{B}\mathbf{M}^T\mathbf{H}^T\right]^{-1}\mathbf{d}$, intégrées de manières rétrogrades jusqu'à l'instant initial via $\mathbf{M}^T\mathbf{H}^T$. La correction apportée par l'algorithme 4DVar est à l'image de la matrice de covariance d'erreur d'ébauche $\mathbf{B}$: une mauvaise modélisation de celle-ci pourra entraîner une mauvaise correction de la condition initiale.

Formulation duale

Bien souvent, le nombre d'observations est largement inférieur à la dimension de la variable d'état [28], et une minimisation dans cet espace s'avère moins coûteuse. Pour ce faire, on s'intéresse à la formulation duale du problème de minimisation du 4DVar (encore appelée PSAS pour Physical-Space Statistical Analysis System) : posons $\mathbf{s}=\mathbf{y}-\mathbf{H}\mathbf{x}$ et introduisons le lagrangien $\mathcal{L}(\mathbf{x},\mathbf{s},\boldsymbol{\lambda})=\frac{1}{2}(\mathbf{x}-\mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}-\mathbf{x}^b)+\frac{1}{2}s^T\mathbf{R}^{-1}\mathbf{s}+\boldsymbol{\lambda}^T(\mathbf{y}-\mathbf{H}\mathbf{x}-\mathbf{s})$. La théorie de la dualité [30] amène à maximiser

$$q(\boldsymbol{\lambda})=\inf_{\mathbf{x},\mathbf{s}}\mathcal{L}(\mathbf{x},\mathbf{s},\boldsymbol{\lambda}).$$

Or $\inf_{\mathbf{x},\mathbf{s}}\mathcal{L}(\mathbf{x},\mathbf{s},\boldsymbol{\lambda})$ est atteint en $(\bar{\mathbf{x}},\bar{\mathbf{s}})$ tel que

$$\begin{aligned}\nabla_{\mathbf{x}}\mathcal{L}(\bar{\mathbf{x}},\bar{\mathbf{s}},\boldsymbol{\lambda})&=0\\ \nabla_{\mathbf{s}}\mathcal{L}(\bar{\mathbf{x}},\bar{\mathbf{s}},\boldsymbol{\lambda})&=0,\end{aligned}$$

soit lorsque $\bar{\mathbf{s}}=\mathbf{R}\boldsymbol{\lambda}$ et $\bar{\mathbf{x}}=\mathbf{x}^b+\mathbf{B}\mathbf{H}^T\boldsymbol{\lambda}$. En réinjectant dans l'expression de $\mathcal{L}$ on cherche donc

$$\max_{\boldsymbol{\lambda}}\mathcal{L}(\bar{\mathbf{x}},\bar{\mathbf{s}},\boldsymbol{\lambda})=\frac{1}{2}\boldsymbol{\lambda}^T(\mathbf{R}+\mathbf{H}\mathbf{B}\mathbf{H}^T)\boldsymbol{\lambda}-\boldsymbol{\lambda}^T(\mathbf{y}-\mathbf{H}\mathbf{x}^b). \tag{1.20}$$

Les deux formalismes ont été reliés par P. Courtier [29] qui a montré qu'elles partageaient le même conditionnement. Au premier abord, la formulation duale a le désavantage que des itérations successives menant à la résolution de (1.20) ne s'accompagnent pas nécessairement d'une décroissance monotone de la fonctionnelle primale $\Omega(x)$, il est donc difficile de savoir quand arrêter ces itérations. Toutefois, nous verrons à la section 2.1.5 une manière d'exploiter la dimension inférieure de l'espace des observations tout en ayant les mêmes itérations que celles obtenues par une minimisation du problème primal.

4DVar à contraintes faibles

Les formulations du 4DVar présentées jusqu'à maintenant sont dites à contraintes fortes : on suppose le modèle parfait, c'est à dire qu'il n'y pas d'erreur lors de la propagation de l'état à l'instant t_i à l'état à l'instant t_{i+1}. On peut incorporer une erreur modèle dans la fonctionnelle du 4DVar dite à contraintes faibles [65] :

$$\Omega(\mathbf{x}) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=1}^{N}(\mathbf{x}_i - \mathcal{M}_i(\mathbf{x}_{i-1}))^T\mathbf{Q}_i^{-1}(\mathbf{x}_i - \mathcal{M}_i(\mathbf{x}_{i-1})) \tag{1.21}$$
$$+ \frac{1}{2}\sum_{i=0}^{N}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i)^T\mathbf{R}_i^{-1}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i)$$

avec $\mathbf{Q}_i$ la matrice de covariance d'erreur modèle (voir (1.3)). Cependant [66] souligne que la matrice $\mathbf{Q}$ n'est pas modélisable facilement, et que l'optimisation portant sur l'ensemble des vecteurs d'états pendant la fenêtre d'assimilation $(\mathbf{x}_0, \mathbf{x}_1, ..., \mathbf{x}_N)$ rend le calcul excessivement coûteux.

Il est possible de réduire la taille du vecteur de contrôle en divisant la fenêtre d'assimilation en m intervalles et en ne considérant le terme d'écart au modèle qu'au début de ces intervalles. Si le début du $k_i^{\text{ème}}$ intervalle $(i = 1, 2, ..., m)$ comporte p pas de temps, le 4DVar à contraintes faibles modifié s'écrit

$$\begin{aligned}\Omega(\mathbf{x}_{k_1}, ..., \mathbf{x}_{k_m}) = &\frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) \\ &+ \frac{1}{2}\sum_{i=1}^{m}(\mathbf{x}_{k_i} - \mathcal{M}_{k_i}^{p}(\mathbf{x}_{k_{i-1}}))^T\mathbf{Q}_{k_i}^{-1}(\mathbf{x}_{k_i} - \mathcal{M}_{k_i}^{p}(\mathbf{x}_{k_{i-1}})) \\ &+ \frac{1}{2}\sum_{i=0}^{m}\sum_{j=0}^{p-1}(\mathcal{H}_{k_i+j}[\mathcal{M}_{k_i}^{j}(\mathbf{x}_{k_i})] - \mathbf{y}_{k_i+j})^T\mathbf{R}_{k_i+j}^{-1}\mathcal{H}_{k_i+j}[\mathcal{M}_{k_i}^{j}(\mathbf{x}_{k_i})] - \mathbf{y}_{k_i+j})\end{aligned} \tag{1.22}$$

où l'opérateur non linéaire discret $\mathcal{M}_i^j$ intègre le modèle sur j pas de temps à partir de l'instant t_i.

Toujours dans [66], Trémolet propose de diviser les fenêtres d'assimilation en plus petites fenêtres pour supposer que l'erreur modèle est constante sur ces intervalles de temps, mais peut varier d'un intervalle à l'autre. En faisant l'hypothèse que $\mathbf{x}_i = \mathcal{M}_{0,i}(\mathbf{x}_0) + \boldsymbol{\eta}_i$ avec $\boldsymbol{\eta}_i$ le vecteur aléatoire représentant l'erreur modèle sur le $i^{\text{ème}}$ intervalle, on est amené à minimiser la fonctionnelle forcée

$$\begin{aligned}\Omega(\mathbf{x}_0, \boldsymbol{\eta}) = &\frac{1}{2}(\mathbf{x}_0 - \mathbf{x}^b)^T\mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}^b) + \frac{1}{2}\sum_{i=1}^{N}\boldsymbol{\eta}_i^T\mathbf{Q}_i^{-1}\boldsymbol{\eta}_i \\ &+ \frac{1}{2}\sum_{i=0}^{N}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i)^T\mathbf{R}_i^{-1}(\mathcal{H}_i[\mathcal{M}_{0,i}(\mathbf{x}_0)] - \mathbf{y}_i).\end{aligned} \tag{1.23}$$

1.3 Régularisation et assimilation de données

Comme précédemment mentionné, le nombre d'observations est souvent très inférieur à la dimension de la variable d'état, donnant lieu à un problème mal posé au sens où il peut exister une infinité de solutions. La régularisation permet d'en sélectionner une particulière ; c'est une manière d'injecter de l'information a priori. Nous dressons à présent un panorama des diverses formes de régularisations possibles et leurs particularités.

1.3.1 Le 4DVar comme une régularisation de Tikhonov des observations

Encore appelée Ridge Regression, la régularisation de Tikhonov consiste en la pénalisation d'un problème aux moindres carrés avec une norme L_2 portant sur la variable d'intérêt ou sur une transformation linéaire de celle-ci [31]. Elle s'écrit donc d'une manière générale comme

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\Gamma(\mathbf{x} - \mathbf{x}_0)\|_2^2$$

Le terme d'écart à l'ébauche $\frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} = \frac{1}{2}\|\mathbf{B}^{-1/2}(\mathbf{x}_0 - \mathbf{x}^b)\|_2^2$ de la fonctionnelle du 4DVar (1.6) peut ainsi être vu comme un tel terme de régularisation.

Remarquons qu'un algorithme capable de minimiser $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$ pourra minimiser $\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\Gamma(\mathbf{x} - \mathbf{x}_0)\|_2^2 = \| \begin{pmatrix} \mathbf{A} \\ \sqrt{\lambda}\Gamma \end{pmatrix} \mathbf{x} - \begin{pmatrix} b \\ \sqrt{\lambda}\Gamma\mathbf{x}_0 \end{pmatrix} \|_2^2$ similairement. Dans le cas où $\Gamma = \mathbf{I}$ et $\mathbf{x}_0 = 0$, la solution du problème aux moindres carrés correspondant s'écrit donc $\hat{x} = (\mathbf{A}^T\mathbf{A} + \lambda\Gamma^T\Gamma)^{-1}\mathbf{A}^T\mathbf{b} = \sum_j \frac{\sigma_j^2}{\sigma_j^2+\lambda}\frac{\mathbf{u}_j^T\mathbf{b}}{\sigma_j}\mathbf{v}_j$ avec $\mathbf{u}_j$ et $\mathbf{v}_j$ les vecteurs singuliers de $\mathbf{A}$ associés aux valeurs singulières σ_j. Le paramètre λ va donc filtrer les plus petites valeurs singulières lorsque $\lambda \gg \sigma_j$.

Prendre $\Gamma = \mathbf{I}$ va promouvoir un écart à l'ébauche de norme faible. Il est également possible de prendre pour Γ un opérateur de différence finie discrétisé, ou une matrice de passage pour pénaliser $\mathbf{x}$ dans une certaine base. Dans tous les cas, l'objectif est de pénaliser $\mathbf{x}$ dans une base où ce vecteur est « parcimonieux », ou « creux », c'est à dire que la majorité de ses composantes sont nulles. Une base des coefficients de Fourier ou une base d'ondelettes (Daubechies, Haar ...) ont montré leur pertinence dans de nombreuses applications [32]. Attention cependant, une mauvaise base risque de détériorer la stabilité numérique de la reconstruction du signal (en particulier pour les bases non orthonormales).

Quant au paramètre de régularisation λ, celui-ci va peser à quel point la régularisation va jouer dans la minimisation. Nous verrons plus tard plusieurs manières de le déterminer, soit théoriquement soit expérimentalement. Une autre manière de voir le lien entre le 4Dvar et la régularisation de Tikhonov, proposée dans [33], est de poser $\mathbf{C}_B = \frac{1}{\sigma_b^2}\mathbf{B}$,

$\mathbf{C}_R = \frac{1}{\sigma_o^2}\mathbf{R}$ et d'effectuer le changement de variable $\mathbf{z} = \mathbf{C}_B^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^b)$ dans (1.6), menant à

$$\Omega(\mathbf{z}) = \|\mathbf{C}_R^{-1/2}(\mathbf{y} - \hat{\mathcal{H}}(\mathbf{x}_0^b)) - \mathbf{C}_R^{-1/2}\hat{\mathcal{H}}\mathbf{C}_B^{1/2}\mathbf{z}\|_2^2 + \mu\|\mathbf{z}\|_2^2 \tag{1.24}$$

avec $\mu^2 = \frac{\sigma_o^2}{\sigma_b^2}$ le paramètre de régularisation. À nouveau, l'opérateur $\mathbf{G} = \mathbf{C}_R^{-1/2}\hat{\mathcal{H}}\mathbf{C}_B^{1/2}$ peut être mal conditionné. Toujours dans [33], les auteurs vont ajouter à (1.24) une autre régularisation en norme 1, qui correspond donc à une pénalisation « mixte » en norme L_1/L_2 :

$$\Omega(\mathbf{z}) = \|\mathbf{C}_R^{-1/2}(\mathbf{y} - \hat{\mathcal{H}}(\mathbf{x}_0^b)) - \mathbf{C}_R^{-1/2}\hat{\mathcal{H}}\mathbf{C}_B^{1/2}\mathbf{z}\|_2^2 + \mu\|\mathbf{z}\|_2^2 + \lambda\|\mathbf{z}\|_1.$$

Nous regardons alors les avantages et inconvénient de l'ajout d'un terme de régularisation en norme L_1, référencé dans la littérature sous le nom de régularisation LASSO (Least Absolute Shrinkage and Selection Operator).

1.3.2 Les régularisations LASSO, Elastic Net et Fused-LASSO

Cette régression amène à considérer le problème

$$\begin{aligned} \min_x \quad & \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 \\ \text{s.t.} \quad & \|\mathbf{x}\|_1 \leq t \end{aligned} \tag{1.25}$$

avec $t > 0$ fixé, ou, sous sa forme lagrangienne :

$$\min_x \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|_1. \tag{1.26}$$

Par rapport à la régularisation de Tikhonov, la forme (1.26) montre que cette régularisation va pénaliser plus fortement les composantes entre 0 et 1 de $\mathbf{x}$. De même, à cause de la forme carrée des lignes de niveau de la norme 1, cette régularisation est plus à même de mettre à 0 les faibles composantes par rapport à celle de Tikhonov. La norme 1 est en fait la relaxation convexe de la norme l_0 définit par $l_0(\mathbf{x}) =$ « nombre de composantes non nulles de $\mathbf{x}$ ».

La fonctionnelle pénalisée en norme L_1 (1.25) n'est plus différentiable. On peut soit avoir recours à des algorithmes d'optimisation non lisse, soit à des algorithmes d'optimisation avec contraintes, comme proposé dans [33] : on commence par séparer $\mathbf{x}$ en parties positives et négatives $\mathbf{x} = \mathbf{x}^+ - \mathbf{x}^-$, avec $\mathbf{x}^+ = \max(\mathbf{x}, 0)$ et $\mathbf{x}^- = \max(-\mathbf{x}, 0)$ (à comprendre composante par composante). Le problème se réécrit alors

$$\begin{aligned} \min_{\mathbf{x},\mathbf{x}^+,\mathbf{x}^-} \quad & \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \mathbf{1}^T\mathbf{x}^+ + \mathbf{1}^T\mathbf{x}^- \\ \text{s.t.} \quad & \mathbf{x} = \mathbf{x}^+ - \mathbf{x}^- \\ & \mathbf{x}^+, \mathbf{x}^- \geq 0 \end{aligned} \tag{1.27}$$

avec $\mathbf{1}$ le vecteur ne comportant que des 1 et de taille appropriée. Ce problème peut à nouveau être écrit sous la forme d'un problème quadratique :

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}\mathbf{w}^T\mathbf{S}\mathbf{w} + \mathbf{c}^T\mathbf{w} \\ \text{s.t.} \quad & \mathbf{E}\mathbf{w} = \mathbf{0} \\ & \mathbf{F}\mathbf{w} \geq 0 \end{aligned} \tag{1.28}$$

(1.29)

avec $\mathbf{w} = \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^+ \\ \mathbf{x}^- \end{bmatrix}$, $\mathbf{S} = \begin{bmatrix} \mathbf{A}^T\mathbf{A} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$, $\mathbf{c} = \begin{bmatrix} -2\mathbf{A}^T\mathbf{b} \\ \mathbf{1} \\ \mathbf{1} \end{bmatrix}$, $\mathbf{F} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix}$ et $\mathbf{E} = \begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \\ \mathbf{I} \end{bmatrix}$.

où les blocs $\mathbf{0}$ et $\mathbf{I}$ sont de taille consistante.

Lorsqu'un champs possède de nombreux groupes de variables très corrélées, le LASSO va avoir tendance à sélectionner une variable parmi un groupe et ignorer les autres [36]. La technique de « l'Elastic Net » a été proposée en statistique pour atténuer cet effet en combinant les deux régularisations vues précédemment :

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\|\mathbf{x}\|_2^2.$$

Il s'agit donc en fait de la régularisation mixte L_1/L_2 proposée par Freitag *et al.* pour l'assimilation de données. Enfin, le Fused-LASSO [37] va également pénaliser les variations de $\mathbf{x}$ en ajoutant à la minimisation le terme $\mathbf{D}\mathbf{x}$:

$$\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2 + \lambda_1\|\mathbf{x}\|_1 + \lambda_2\|\mathbf{D}\mathbf{x}\|_1. \tag{1.30}$$

avec

$$\mathbf{D} = \begin{bmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix} \tag{1.31}$$

de sorte que $\|\mathbf{D}\mathbf{x}\|_1 = \sum_{j=1}^{n-1}|\mathbf{x}_{j+1} - \mathbf{x}_j|$. Le second terme de (1.30) va alors pénaliser les changements brusques des magnitudes des composantes.

Les algorithmes de pointe utilisés pour faire décroître des fonctions régulières pénalisées par ce type de normes font appel aux méthodes proximales rappelées dans l'annexe (B). Pour le problème LASSO plus particulièrement, une méthode d'ordre 2 pour les grands systèmes a récemment été proposée dans [38] et est basée sur les itérations de « semismooth Newton », qui généralisent les itérations de Newton (rappelées en section (2.1.3)) pour des fonctions qui peuvent être seulement localement lipschitzienne.

La norme d'Huber ([39]) est une autre manière de mêler pénalisation en norme L_1 et L_2 : cette norme est quadratique sur une boule de rayon δ centrée en 0 et devient linéaire en dehors de cette boule :

$$\|x\|_\delta^{\text{Huber}} = \begin{cases} \frac{1}{2}x^2 & \text{si } |x| \leq \delta, \\ \delta \cdot \left(|x| - \frac{1}{2}\delta\right), & \text{sinon.} \end{cases}$$

Ayant considéré des pénalisations avec des normes L_1 et L_2, il est naturel de s'intéresser à une pénalisation plus générale en norme L_p avec p quelconque.

1.3.3 La régularisation en norme L_p

Commençons par remarquer que pour le cas $p = 0$, l_0 ne vérifie pas la propriété $l_0(\mathbf{x}) = 0 \Rightarrow \mathbf{x} = 0$. Bien que cette régularisation permet de pénaliser fortement les composantes non nulles, cette fonction n'est pas différentiable. Des méthodes pour minimiser la fonctionnelle pénalisée dans ce cas s'appuient sur des approximations polynomiales successives comme [40], ou par exemple sur une relaxation continue de la nature discrète de cette norme [41].

Pour $0 < p < 1$, la fonction $\|\cdot\|_p^p$ est différentiable mais n'est pas convexe. Aussi, il ne s'agit toujours pas d'une norme puisqu'elle ne vérifie pas l'inégalité triangulaire, mais d'une quasi-norme. Une telle régularisation a été envisagée par exemple pour l'optimisation de matrices de permutations [42] et la compression/restauration d'images ([43]). Sa relation vis-à-vis de la sparsité de la solution a été étudiée dans [44]. Dans ce dernier papier, les auteurs minimisent la fonctionnelle (continue mais non convexe, ni lipschitzienne) pénalisée via une méthode de hybrid orthogonal matching pursuit-smoothing gradient.

La norme L_p, dont l'expression est rappelée à la définition (2.2.3), est bien une norme pour $p > 1$. Elle a plusieurs intérêts. D'abord, même dans le cas d'un signal parcimonieux où une régularisation en norme L_1 serait attendue, Schuster *et al.* [48] suggèrent d'utiliser la norme L_p avec p légèrement supérieur à 1 ce qui permet d'atténuer les oscillations de Gibbs surgissant lors de la reconstruction d'un signal présentant des discontinuités. Suite à ces remarques, [45] a montré, toujours sur un problème de reconstruction de signal, que la norme p avec $1 < p < 2$ amenait à de meilleures performances dans le cas de

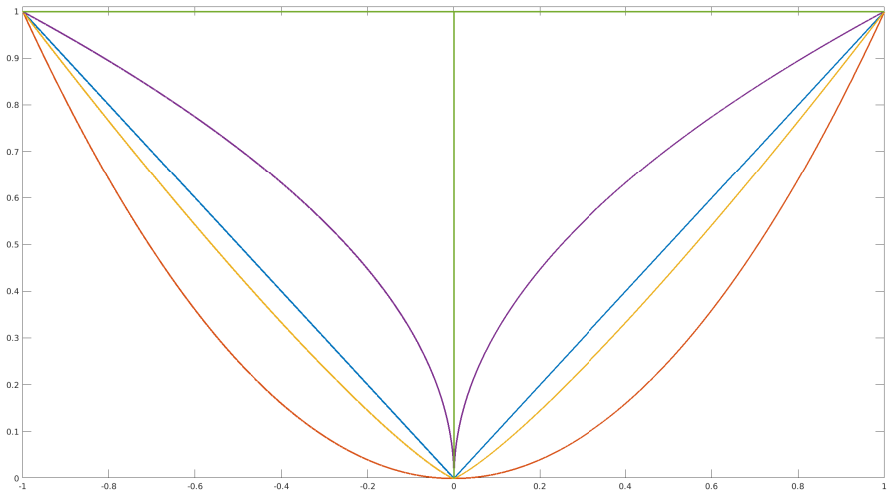


FIGURE 1.3 – Représentation graphique des pénalisations $t \to t^p$ pour différents p en une dimension : l_0 (vert), $l_{0.5}$ (violet), l_1 (bleu), $l_{1.2}$ (jaune) et l_2 (orange).

discontinuités très localisées (le signal était une fonction porte de faible largeur). Nous espérons retrouver les mêmes avantages sur un problème d'assimilation de données qui s'écrira donc sous une forme générale comme un problème aux moindres carrés non linéaire pénalisé :

$$\|\mathbf{A}(\mathbf{x}) - \mathbf{b}\|_2^2 + \lambda \|\mathbf{\Phi x}\|_p^p, \tag{1.32}$$

avec $\mathbf{\Phi}$ une matrice de passage pour effectuer la pénalisation dans une base parcimonieuse pour $\mathbf{x}$. La norme p étant élevée à la puissance p, le terme de pénalisation est infiniment différentiable. Une représentation graphique des pénalisations des normes L_p pour différents p est donnée figure (1.3).

Dans cette thèse, nous avançons au cours des chapitres d'autres arguments pour l'utilisation de cette norme. Nous montrerons en effet qu'elle est particulièrement pertinente pour des signaux « quasi-creux », et aussi qu'elle peut être justifiée théoriquement selon les densités de probabilité suivies par les variables lors de la modélisation du problème.

1.3.4 Choix des paramètres de régularisation

Un des principaux défis lors de l'utilisation d'un terme de pénalisation est le choix du paramètre λ qui se base nécessairement sur une heuristique. Lorsque λ tend vers zéro, on se ramène à la fonctionnelle du 4DVar classique, tandis que le cas où λ tend vers l'infini sélectionne le minimum de $\|\mathbf{\Phi x}\|_p^p$, autrement dit le vecteur nul. Un paramètre non nul diminuera l'influence des termes d'écart aux observations et d'écart à l'ébauche mais sélectionnera avec espoir une solution avec de meilleurs propriétés.

Deux heuristiques très populaires pour le choix de λ sont la méthode de la « L-curve »[46] et le principe de Morozov [47].

Principe de Morozov

On cherche ici à résoudre $\mathbf{Ax} = \mathbf{b}_{no-noise} + \boldsymbol{\varepsilon}$: on a seulement accès à des données bruitées par un terme $\boldsymbol{\varepsilon}$. On prend en compte cette incertitude sur le second membre en introduisant une régularisation sur $\mathbf{x}$ en norme $\|\cdot\|$ quelconque :

$$\min_{\mathbf{x}} \Omega(\mathbf{x}, \mathbf{b}, \lambda) = \|\mathbf{Ax} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{x}\|. \tag{1.33}$$

Si on dispose d'une estimation de l'ordre de grandeur du bruit : $\|\boldsymbol{\varepsilon}\| < \delta$, on peut considérer que $\bar{\mathbf{x}}$ est une reconstruction acceptable si

$$\|\mathbf{A}\bar{\mathbf{x}} - \mathbf{b}\| \leq \delta.$$

Dans le cas d'une norme L_2 et lorsque $\boldsymbol{\varepsilon}$ est un bruit gaussien dont les n composantes sont $\varepsilon^j \sim \mathcal{N}(0, \sigma^2)$, on peut prendre $\delta = \sqrt{n}\sigma$ puisque l'espérance de l'erreur est $\mathbb{E}(\|\boldsymbol{\varepsilon}\|) = \sqrt{n}\sigma$. De même, une solution $\bar{\mathbf{x}}$ qui minimiserait uniquement le terme $\|\mathbf{Ax} - \mathbf{b}\|$ et ignorerait la régularisation ne serait pas souhaitable. Le principe de Morozov s'énonce alors :

Définition 1.3.1 (Principe de Morozov / Morozov's discrepancy principle). *Soit $\tau_2 \geq \tau_1 \geq 1$ et $\mathbf{b}_{no-noise} \in Im(\mathbf{A})$ les données non bruitées. Pour $\delta > 0$ et $\mathbf{b}$ tel que $\|\mathbf{b} - \mathbf{b}_{no-noise}\| \leq \delta$, on dit que le paramètre de régularisation $\lambda > 0$ est choisi d'après le principe de Morozov s'il existe $\mathbf{x}_\lambda^\delta$ tel que :*

$$\mathbf{x}_\lambda^\delta = \arg\min_{\mathbf{x}} \Omega(\mathbf{x}, \mathbf{b}, \lambda) \tag{1.34}$$

$$et\ \tau_1\delta \leq \|\mathbf{Ax}_\lambda^\delta - \mathbf{b}\| \leq \tau_2\delta. \tag{1.35}$$

Pour trouver une valeur de λ en accord avec le principe de Morozov, une séquence de minima de Ω, $\left\{\mathbf{x}_\lambda^\delta\right\}_\lambda$, dépendant de λ, doit être calculée jusqu'à ce que les inégalités (1.35) soient satisfaites. Le choix des paramètres λ pour trouver une solution $\mathbf{x}_\lambda^\delta$ respectant (1.35) s'appuie sur la propriété de monotonie de la fonction $\lambda \to \|\mathbf{Ax}_\lambda^\delta - \mathbf{b}\|$, qui permet en pratique d'utiliser un procédé de dichotomie.

Méthode de la L-curve

La méthode dite de la L-curve pour choisir λ repose sur la volonté de réaliser un compromis entre les deux termes à minimiser (l'écart au second membre bruité et la régularisation).

À nouveau une séquence de paramètres

$$0 < \lambda_1 < \lambda_2 < ... < \lambda_M < \infty$$

est choisie et le minimum $\bar{\mathbf{x}}_{\lambda_j}$ du problème régularisé est calculé pour $1 \leq j \leq M$. On trace alors la courbe

$$(\log(\|\mathbf{A}\bar{\mathbf{x}}_{\lambda_j} - \mathbf{b}\|), \log(\|\mathbf{x}\|)$$

qui peut avoir l'allure d'une courbe « en L ». La valeur de λ optimale est alors choisie dans le creux de la courbe, comme indiqué sur la Figure (1.4). C'est la valeur qui, pour de petites variations de λ, défavorise le moins chacun des deux termes concurrents. Malheureusement une courbe lisse nécessite beaucoup de points à calculer et son allure peut ne pas être claire, rendant le choix de λ difficile.

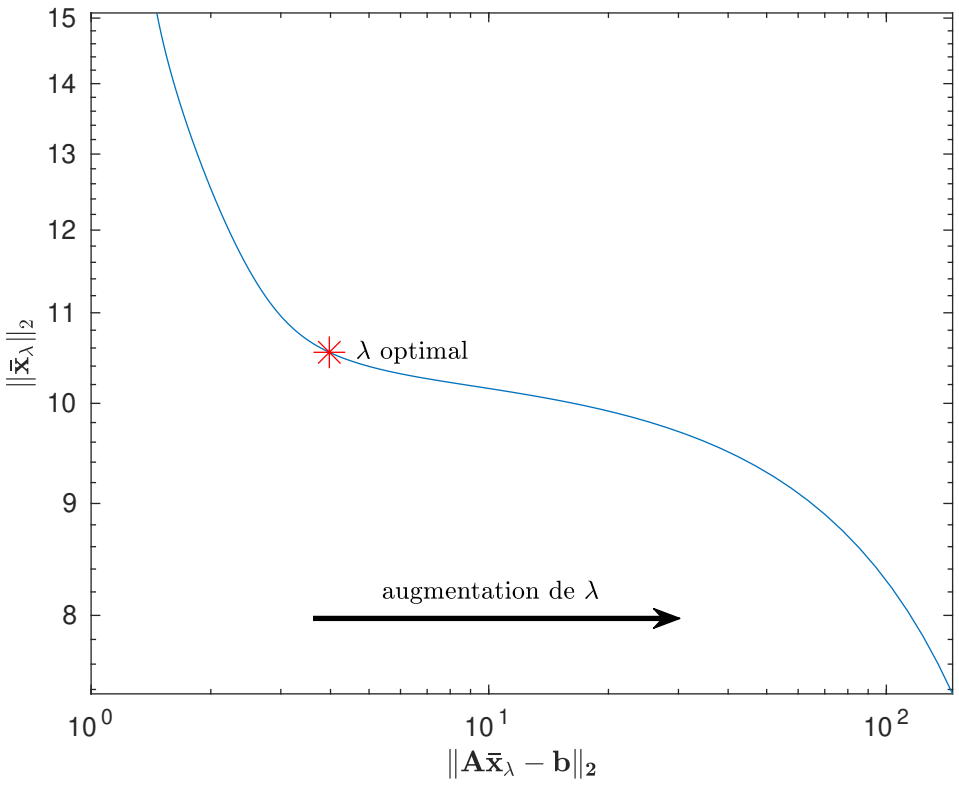


FIGURE 1.4 – Choix du paramètre de régularisation optimal par la méthode de la L-curve (dans un repère log-log). Ici, la norme utilisée est la norme L_2, $\mathbf{A}$ est une matrice aléatoire de taille 100x100 et $\mathbf{b}$ est bruité par un bruit gaussien $\mathcal{N}(\mathbf{0}, \sqrt{3}\mathbf{I})$.

Conclusion

Nous avons vu qu'au centre de l'assimilation de données variationnelle était la minimisation d'une fonctionnelle, en général non linéaire. De plus, l'ajout d'un terme de régularisation est souvent pertinent dans ce contexte, où les variables d'états sont de dimension très élevée par rapport au nombre d'observations, pour sélectionner une solution particulière. Le choix de la norme utilisée pour la pénalisation dépend du problème et va avoir un impact sur la facilité à effectuer la minimisation de la fonctionnelle. Il s'agit à présent de présenter quels sont les algorithmes utilisés à cette fin.

Chapitre 2

Optimisation et assimilation de données : algorithmes de minimisation du 4DVar

2.1 Minimisation dans les espaces de Hilbert 42
2.1.1 Algorithmes de descente : généralités 42
2.1.2 Algorithme de descente de gradient 44
2.1.3 Algorithme de Newton 44
2.1.4 Algorithme de Gauss-Newton 44
2.1.5 Algorithme du gradient conjugué linéaire pour la boucle interne . . 46
2.1.6 Algorithme de Krylov pour la boucle interne 47
2.1.7 Algorithme du gradient conjugué non linéaire 47
2.1.8 Algorithme BFGS 48
2.2 Minimisation dans les espaces de Banach 49
2.2.1 Géométrie des espaces de Banach 49
2.2.2 Descente de gradient dans les espaces de Banach 54
2.2.3 Algorithmes de gradient conjugué dans les espaces de Banach . . . 57
2.3 Conclusion 59

Ce chapitre présente les algorithmes déjà existants utilisés en pratique pour minimiser le 4DVar. Il commence par des rappels d'optimisation qui seront réutilisés au chapitre 4. Toute la théorie classiquement utilisée par ces algorithmes est celle des espaces de Hilbert. Plus méconnu en assimilation de données, le cadre des espaces de Banach s'avère propice à l'étude de la régularisation. En particulier, il va jouer un rôle important dans la conception d'algorithmes efficaces pour minimiser le 4DVar pénalisé par une norme L_p. Nous introduisons donc ce cadre et ces algorithmes dans un second temps.

2.1 Minimisation dans les espaces de Hilbert

L'espace de Hilbert est le cadre classique et le plus répandu en assimilation de données variationnelles. Les points abordés dans cette première section concernent donc l'état de l'art de ce qui est fait actuellement pour minimiser le 4DVar.

2.1.1 Algorithmes de descente : généralités

La fonctionnelle du 4DVar pénalisée s'écrit donc :

$$\Omega(\mathbf{x}_0) = \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \hat{\mathcal{H}}(\mathbf{x}_0)\|^2_{\mathbf{R}^{-1}} + \lambda\|\mathbf{\Phi}\mathbf{x}_0\|^p_p. \tag{2.1}$$

Les algorithmes destinés à minimiser (2.1) ont la forme générale suivante :

$$x_{k+1} = x_k + \alpha_k p_k. \tag{2.2}$$

Il s'agit d'un algorithme de descente si p_k est effectivement une direction de descente, c'est à dire, par définition :

Définition 2.1.1 (Direction de descente). *p est une direction de descente pour f au point x si $\langle \nabla f(x), p\rangle < 0$.*

Ainsi si le **pas** α_k est suffisamment petit, f aura décru en passant de x_k à x_{k+1}. Nous utilisons la convention d'écriture f_k pour désigner l'évaluation de f au point x_k : $f_k = f(x_k)$. L'algorithme converge en demandant au pas de respecter certaines conditions, que nous allons maintenant détailler. Les preuves des propriétés suivantes peuvent être retrouvées dans [30].

Méthodes de recherche linéaire du pas : condition d'Armijo

La condition d'Armijo est une condition de décroissance suffisante pour f. Étant donné $c_1 \in]0;1[$, elle impose :

$$f(x_k + \alpha_k p_k) \leq f_k + c_1\alpha_k p_k^T \nabla f_k, \tag{2.3}$$

soit, graphiquement, que le point x_{k+1} soit tel que f_{k+1} soit en dessous de la pente en 0 de la fonction $g(\alpha) = f(x_k + \alpha_k p_k)$, relevée par le coefficient c_1. Un pas α_k satisfaisant cette condition peut être calculé par backtracking : on se donne un pas initial α_k^0, un coefficient $\rho \in]0;1[$ et on itère sur l'indice i :

$$\alpha_k^{i+1} = \rho\alpha_k^i$$

jusqu'à ce que la condition d'Armijo (2.3) soit vérifiée (ce qui est éventuellement le cas pour α_k suffisamment petit, grâce au fait que p_k est une direction de descente). En prenant le plus petit indice qui rend (2.3) vrai, l'algorithme va converger mais, à elle seule, cette condition risque de fournir un pas très petit et empêcher l'algorithme de progresser rapidement. Une seconde condition va permettre de pallier cet inconvénient.

Méthodes de recherche linéaire du pas : conditions de Wolfe

On rajoute une condition dite de courbure permettant de faire des pas plus grands en se donnant $c_2 \in]c_1; 1[$:

$$p_k^T \nabla f(x_k + \alpha_k p_k) \geq c_2 p_k^T \nabla f(x_k). \tag{2.4}$$

Graphiquement, cette condition impose à la dérivée de g au point α_k d'être au dessus de la pente de g en 0 relevée par c_2. De cette manière, si la pente de g en α_k, $g'(\alpha_k)$, est fortement négative, on peut espérer faire décroître f en augmentant le pas α. Les deux conditions (2.3) et (2.4) réunies sont appelées conditions de Wolfe. Elles peuvent être vérifiées simultanément par exemple par un algorithme de bissection/dichotomie : le pas est réduit jusqu'à ce qu'il vérifie la première condition d'Armijo, puis augmenté jusqu'à vérifier la seconde condition de courbure et ainsi de suite.

L'existence d'un pas vérifiant les conditions de Wolfe est assurée dès que f est lisse et bornée inférieurement (ce qui est notre cas en assimilation de données).

Convergence des algorithmes de descente

Soumis aux conditions de Wolfe, un schéma itératif du type (2.2) a pour conséquence le théorème de Zoutendijk qui permet d'obtenir la convergence de ces algorithmes dès qu'une certaine condition d'angle entre $-\nabla f_k$ et p_k est vérifiée quel que soit k.

Il faut également rajouter quelques hypothèses, en pratique non restrictives dans notre cadre d'application, concernant f. Notamment l'hypothèse que l'on retrouve dans [67] :

Hypothèse 2.1.1. — *(i) L'ensemble de niveau $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$, où x_0 est le point de départ des itérations, est borné.*

— *(ii) La fonction objectif f est continûment différentiable sur un voisinage ouvert $\mathcal{N}$ de $\mathcal{L}$, et son gradient est lipschitzien : il existe $L > 0$ tel que*

$$\forall x, \tilde{x} \in N, \|\nabla f(x) - \nabla f(\tilde{x})\|_2 \leq L\|x - \tilde{x}\|_2.$$

Via cette hypothèse le théorème de Zoutendijk s'énonce :

Théorème 2.1.2 (Théorème de Zoutendijk)**.** *Soit un schéma itératif de la forme $x_{k+1} = x_k + \alpha_k p_k$ où p_k est une direction de descente et α_k respecte les deux conditions de Wolfe (2.3) et (2.4), et où f est de classe C^1 sur un ensemble ouvert $\mathcal{N}$ contenant $\mathcal{L}$. Sous l'hypothèse (2.1.1), et en notant*

$$\cos(\theta_k) = -\frac{\langle \nabla f_k, p_k \rangle}{\|\nabla f_k\|_2 \|p_k\|_2},$$

on a

$$\sum_{k \geq 0} \cos^2(\theta_k) \|\nabla f_k\|_2^2 < \infty.$$

Comme conséquence immédiate, $\cos(\theta_k)\|\nabla f_k\|_2 \underset{k\to+\infty}{\longrightarrow} 0$ et en particulier $\|\nabla f_k\|_2 \underset{k\to+\infty}{\longrightarrow} 0$ dès que le cosinus est borné inférieurement par une constante strictement positive, c'est à dire $\cos(\theta_k) \geq \varepsilon > 0$ pour $k \in \mathbb{N}$ suffisamment grand.

2.1.2 Algorithme de descente de gradient

Puisque 2.1 est une fonctionnelle lisse, le premier algorithme de descente auquel on peut penser est celui de la descente de gradient : $p_k = -\nabla f_k$. Il s'agit non seulement bien d'une direction de descente évidente, mais également d'une direction de plus forte pente de f en x_k. Sa convergence grâce au théorème de Zoutendijk est immédiate puisqu'alors pour tout k, $cos(\theta_k) = 1$. Il s'agit d'un algorithme d'ordre 1 : seules des informations du premier ordre (la dérivée première) sont utilisées.

L'algorithme de Newton permet d'exploiter des informations du second ordre (la hessienne de f) pour converger plus rapidement.

2.1.3 Algorithme de Newton

L'algorithme de Newton est initialement mis en oeuvre pour trouver un zéro d'une fonction de classe C^2 non linéaire, en approximant localement la fonction par sa dérivée. Appliqué à la dérivée de la fonction, il permet de trouver un point critique de f. Dans le cas d'une fonction localement convexe, cette méthode est alors équivalente à approximer localement f autour de x_k par un modèle quadratique m_k (son expansion de Taylor à l'ordre 2) :

$$m_k(x_k) = f_k + x_k^T \nabla f_k + \frac{1}{2} x_k^T \nabla^2 f_k x_k \tag{2.5}$$

et à résoudre $\nabla m_k(x_k) = 0$. La mise à jour de l'itéré s'effectue alors de la même manière que (2.2) en prenant cette fois pour direction :

$$p_k = -(\nabla^2 f_k)^{-1} \nabla f_k.$$

Cependant le calcul de la hessienne de f peut être trop coûteux pour des applications réelles d'assimilation de données. Une solution est d'approximer la hessienne par une matrice plus simple à calculer et permettant un calcul de $(\nabla^2 f_k)^{-1}\nabla f_k$ plus rapide, qui donne lieu aux méthodes de quasi-Newton.

2.1.4 Algorithme de Gauss-Newton

La première approche possible est celle de Gauss-Newton qui, pour un moindre carré linéaire de la forme $\min f(\mathbf{x}) = \frac{1}{2}\|r(\mathbf{x})\|_2^2$, avec $r(\mathbf{x}) = \begin{pmatrix} r_1(\mathbf{x}) \\ r_2(\mathbf{x}) \\ \vdots \\ r_m(\mathbf{x}) \end{pmatrix}$, va négliger les termes

d'ordre deux dans le calcul de la hessienne : on prendra $\nabla^2 f(x) \approx J_f(\mathbf{x})^T J_f(\mathbf{x})$ (au lieu de sa valeur exacte dont le terme i,j est $\nabla^2 f(\mathbf{x})_{ij} = \sum_{i=1}^{m} \left(\frac{\partial r_i}{\partial \beta_j} \frac{\partial r_i}{\partial \beta_k} + r_i \frac{\partial^2 r_i}{\partial \beta_j \partial \beta_k} \right)$). De manière équivalente, cette méthode itérative recherche un incrément $\delta\mathbf{x}$ tel que $\mathbf{x}_{k+1} = \mathbf{x}_k + \delta\mathbf{x}$ avec $\delta\mathbf{x}$ vérifiant les équations normales $J_f(\mathbf{x})^T J_f(\mathbf{x})\delta\mathbf{x} = J_f(\mathbf{x})^T r(\mathbf{x})$. Dans notre cas, une itération de Gauss-Newton (sans considérer de pénalisation) est en fait équivalente à linéariser les opérateurs non linéaires du 4DVar et trouver l'incrément $\delta\mathbf{x}$ qui minimisera à l'itération k

$$\Omega(\delta\mathbf{x}) = \frac{1}{2}(\mathbf{x}_k + \delta\mathbf{x} - \mathbf{x}_B)\mathbf{B}^{-1}(\mathbf{x}_k + \delta\mathbf{x} - \mathbf{x}_b) + \frac{1}{2}(\hat{\mathbf{H}}\delta\mathbf{x} - \mathbf{d})^T\mathbf{R}^{-1}(\hat{\mathbf{H}}\delta\mathbf{x} - \mathbf{d}), \quad (2.6)$$

qui correspond à la fonctionnelle du 4DVar incrémental [21] présentée section (1.2.2). Si la jacobienne de Ω est de rang plein, la solution de (2.6) qui correspond à une « boucle externe » s'exprime par :

$$\delta\mathbf{x} = (\mathbf{B}^{-1} + \hat{\mathbf{H}}\mathbf{R}^{-1}\hat{\mathbf{H}})^{-1}(\mathbf{B}^{-1}(\mathbf{x}_b - \mathbf{x}_k) + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\mathbf{d}). \quad (2.7)$$

Il est maintenant nécessaire de calculer le second membre de (2.7). En remarquant que $\mathbf{B}^{-1} + \hat{\mathbf{H}}\mathbf{R}^{-1}\hat{\mathbf{H}}$ est définie positive, l'algorithme du gradient conjugué semble bien adapté à résoudre ce problème - qui correspond au problème de la « boucle interne ». En pratique il peut être impossible de stocker les matrices $\hat{\mathbf{H}}$, $\mathbf{B}$ ou $\mathbf{R}$ du fait de leur taille, et seul l'accès au produit de ces matrices par un vecteur peut être disponible. Une méthode de Krylov [53] à mémoire limitée peut alors être pertinente pour résoudre le système linéaire nécessaire au calcul de $\delta\mathbf{x}$.

La convergence de cet algorithme est sensible au point de départ des itérations [22]. Cette sensibilité peut être atténuée en incorporant un terme de pénalisation en norme L_2, $\frac{\lambda_k}{2}\|\mathbf{x}_{k+1}\|_2^2$ à la fonctionnelle linéarisée (2.6). On se ramène en fait à une méthode de Gauss-Newton « amortie », ou algorithme de Levenberg-Marquardt ([54, 55]), qui permet d'une part de rendre la jacobienne J_f de rang plein, et d'autre part de contrôler la taille de l'incrément via le paramètre de régularisation. Cette régularisation du $4DVar$ incrémental linéarisé est à différencier d'une éventuelle régularisation que l'on rajouterait dans la fonctionnelle du $4DVar$ non linéarisée (ce que nous ferons plus tard). Remarquons également que seules des pénalisations en norme L_2 sont ici considérées.

Une manière de rendre l'algorithme global, c'est à dire convergeant quel que soit le point de départ, est d'utiliser une méthode de région de confiance [56]. Ces dernières considèrent successivement des domaines de taille variable sur lesquels la fonctionnelle est approximée par un modèle simple à minimiser.

En résumé, l'algorithme de Gauss-Newton consiste à linéariser $\mathcal{M}$ et $\mathcal{H}$ dans une boucle externe (sans considérer de régularisation) puis résoudre le système (2.7) dans une boucle interne (dans laquelle est éventuellement rajoutée une régularisation). Nous détaillons maintenant la résolution du problème de la boucle interne.

2.1.5 Algorithme du gradient conjugué linéaire pour la boucle interne

L'algorithme du gradient conjugué est d'abord conçu pour minimiser des quadratiques strictement convexes du type $\phi(x) = \frac{1}{2}x^T Ax + b^T x$ avec A définie positive, comme c'est le cas ici. Il peut se voir comme un algorithme de résolution direct qui converge théoriquement en au plus n itérations vers la solution du système, bien qu'en arithmétique finie ce ne sera pas nécessairement le cas [57]. Il peut également se voir comme un algorithme de descente itératif pour lequel le calcul d'une nouvelle direction dépendra de l'ancienne par $p_{k+1} = -\nabla\phi(x_k) + \beta_k p_k$. Une expression close pour β_k est obtenue en imposant à deux directions de descentes successives d'être conjuguées par rapport à la hessienne $\mathbf{H}$ (indépendante de k) de Ω :

$$p_{k+1}^T \mathbf{H} p_k = 0 \tag{2.8}$$

Dans le cas d'une quadratique on a aussi facilement une expression analytique du pas α_k qui va minimiser exactement ϕ selon la droite $x_k + \alpha_k p_k$. En fait, cet algorithme minimise successivement Ω sur l'espace $x_0 + \text{vect}(p_0, p_1, ..., p_{k-1})$. L'algorithme s'écrit :

Algorithme 2 Algorithme du gradient conjugué classique

$r_0 = Ax_0 - b$
$p_0 = -r_0$
$k = 0$
tant que $\|r_k\| \leq \epsilon$ **faire**
    $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$
    $x_{k+1} = x_k + \alpha_k p_k$
    $r_{k+1} = r_k + \alpha_k A p_k$
    $\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
    $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$
    $k = k + 1$
fin tant que

Sa vitesse de convergence dépendra de la répartition des valeurs propres de la matrice A [58], qui peut être améliorée via du préconditionnement [30].

On peut utiliser cet algorithme pour résoudre le sous-problème $(\mathbf{B}^{-1} + \hat{\mathbf{H}}\mathbf{R}^{-1}\hat{\mathbf{H}})^{-1}(\mathbf{B}^{-1}(\mathbf{x}_b - \mathbf{x}_k) + \hat{\mathbf{H}}^T\mathbf{R}^{-1}\mathbf{d})$ dans l'espace des observations (via la formule de Sherman-Morrison), réduisant ainsi la complexité des calculs, tout en assurant que les itérations faites suivront le même chemin que si l'on résolvait directement le système (2.7), permettant d'assurer la décroissance de la fonctionnelle du 4DVar à chaque itération. Ce développement est fait via un changement de produit scalaire astucieux par l'algorithme du Restricted Preconditioned Conjugate Gradient (RPCG) dans [59].

2.1.6 Algorithme de Krylov pour la boucle interne

Cette classe d'algorithme cherche à chaque itération k une approximation de la solution en se restreignant au sous-espace

$$x_0 + \mathcal{K}^k(A, r_0) = x_0 + \text{vect}\,\{r_0, Ar_0, A^2r_0, \ldots, A^{k-1}r_0\}$$

(toujours avec les notations de la section précédente).

L'algorithme de Lanczos, adapté à la résolution de systèmes définis positifs [69], impose notamment la condition dite de Galerkin $r_k \perp \mathcal{K}^k(A, r_0)$, et mène à $V_k^T r_k = 0$ où V_k est une matrice dont les colonnes forment une base orthonormales de $\mathcal{K}^k(A, r_0)$ dont la construction peut être trouvée dans [53]. L'itéré x_k est alors fourni par $x_k = x_0 + V_k y_k$ avec $y_k = (V_k^T A V_k)^{-1}(\|r_0\| e_1)$ avec e_1 le premier vecteur de la base canonique de $\mathbb{R}^n$.

2.1.7 Algorithme du gradient conjugué non linéaire

Parallèlement, l'algorithme du gradient conjugué non linéaire permet de minimiser une fonction f quelconque et ne nécessite donc pas l'imbrication d'une boucle interne dans une boucle externe. La principale différence étant que le résidu r_k qui valait $r_k = Ax_k - b$ dans le cas linéaire, et correspondait au gradient de la quadratique que l'on cherchait à minimiser, sera maintenant égale au gradient de la fonction f (plus forcément quadratique) dans la mise à jour de la direction de descente $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$. Cette généralisation laisse cependant le choix des paramètres α_k et β_k libre. Le pas pourra être à nouveau calculé par une recherche linéaire, tandis que de nombreuses valeurs ont été proposées pour β_k [60]. Se démarquent notamment celle de Fletcher-Rieves qui consiste à remplacer à nouveau r_k par ∇f_k dans le calcul de β_k :

$$\beta_k^{FR} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k} \tag{2.9}$$

ou celle de Hestenes-Stiefel

$$\beta_k^{HS} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{p_k^T (\nabla f_{k+1} - \nabla f_k)} \tag{2.10}$$

Cette valeur vient de l'imposition pour deux directions consécutives d'être conjuguées par rapport à la hessienne moyenne entre x_{k+1} et x_k

$$G_k = \int_0^1 [\nabla^2 f(x_k) + \tau \alpha_k p_k)]\, d\tau,$$

i.e. $p_{k+1}^T G_k p_k = 0$. La direction obtenue $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$ peut ne pas être une direction de descente au point x_k. On peut néanmoins s'y ramener en faisant du backtracking sur β_{k+1} : on le diminue jusqu'à retrouver une direction de descente (ce qui sera le cas éventuellement, puisque $\lim_{\beta_{k+1} \to 0} -\nabla f_{k+1} + \beta_{k+1} p_k = -\nabla f_{k+1}$).

L'algorithme du gradient conjugué non linéaire a la forme suivante :

Algorithme 3 Algorithme du gradient conjugué non-linéaire

1: $p_0 = -r_0$
2: $k = 0$
3: **tant que** $\nabla f_k \neq 0$ **faire**
4: Calcul de α_k
5: $x_{k+1} = x_k + \alpha_k p_k$
6: Calcul de β_k
7: $p_{k+1} = -\nabla f_{k+1} + \beta_{k+1} p_k$
8: $k = k + 1$
9: **fin tant que**

2.1.8 Algorithme BFGS

Nous avons vu comment minimiser le 4DVar (2.5) par l'algorithme du gradient conjugué non linéaire, par la méthode de Newton et par celle de Gauss-Newton (équivalente à la méthode incrémentale), cette dernière correspondant à une méthode de quasi-Newton (elle fait un pas de Newton en utilisant une approximation de la hessienne). Parmi les autres méthodes de quasi-Newton les plus efficaces capables de minimiser Ω, se trouve l'algorithme BFGS de Broyden, Fletcher, Goldfarb et Shanno (conçu et amélioré séparément dans [61], [62], [63], [64]) capable comme le gradient conjugué non linéaire de minimiser une fonction f de classe C^2 pas forcément quadratique. Celui-ci consiste à initialiser la hessienne par $H_0 = \mathbf{I}$ puis à la mettre à jour itérativement. En notant $\mathbf{y}_k = \nabla f_{k+1} - \nabla f_k$ et $\mathbf{B}_k$ l'approximation à l'itération k de $\nabla^2 f_k$, on calcule :

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\alpha_k \mathbf{y}_k^T p_k} - \frac{\mathbf{B}_k p_k p_k^T \mathbf{B}_k^T}{p_k^T \mathbf{B}_k p_k}$$

dont l'inverse peut être calculé efficacement grâce à la formule de Sherman-Morrison :

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\alpha_k p_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k)(p_k p_k^T)}{(p_k^T \mathbf{y}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k p_k^T + p_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{p_k^T \mathbf{y}_k} \tag{2.11}$$

Le problème de l'espace requis pour effectuer la mise à jour précédente (2.11) est atténué par la version Limited Memory BFGS (L-BFGS) proposé par Nocedal [68].

Au centre de notre étude est l'utilisation d'une norme non euclidienne (non issue d'un produit scalaire) comme terme de régularisation. Si certains algorithmes vus précédemment permettent effectivement de minimiser le 4DVar avec une telle pénalisation, le cadre mathématique adéquat est celui des espaces de Banach [48]. Nous allons chercher à nous inspirer des algorithmes de minimisation dans ces espaces particuliers dépourvus de produit scalaire pour minimiser le 4DVar avec une telle pénalisation. À cette fin et dans une optique de comparaison, nous passons à présent en revue certains algorithmes conçus pour effectuer une minimisation dans un espace de Banach.

2.2 Minimisation dans les espaces de Banach

Qu'est-ce qui nous empêche de faire une descente de gradient comme vue à la section (2.1.2) dans ce nouveau cadre mathématique ? Sans produit scalaire, nous n'avons pas à disposition le théorème de Fréchet-Riesz permettant d'identifier la dérivée $f'(x_k)$, qui est par définition un élément du dual topologique de l'espace X où vit la variable x_k, comme un élément de X. Autrement dit le gradient au sens usuel d'une fonction n'est pas bien défini dans ces espaces. Si l'on souhaite faire l'itération (mal posée) $x_{k+1} = \underbrace{x_k}_{\in X} - \alpha_k \underbrace{f'(x_k)}_{\in X^*}$, il va être nécessaire de transporter soit les itérations dans le dual topologique, soit la direction $-f'(x_k)$ dans l'espace primal.

Une étude approfondie de la régularisation dans les espaces de Banach est donné par [48]. Nous commençons par rappeler les outils propres à ces espaces qui seront d'importance pour la suite.

2.2.1 Géométrie des espaces de Banach

Dans ce qui suit X désigne un espace de Banach. Le dual topologique de X est défini ainsi :

Définition 2.2.1 (Espace dual). *On appelle espace dual de X, noté X^*, l'espace de Banach des formes linéaires bornées (continues) $x^* : X \to \mathbb{R}$ équipé de la norme*

$$\|x^*\|_{X^*} = \sup_{\|x\|=1} |x^*(x)|.$$

En particulier, dans toute notre étude, nous désignerons toujours par « espace dual » le dual topologique de l'espace des états **qui n'est pas l'espace des observations du vocabulaire d'assimilation de données**. C'est dans cet espace que s'effectuent les itérations des algorithmes « duaux ».

Définition 2.2.2 (Exposants conjugués). *Soit $p > 1$. L'exposant conjugué de p noté q est défini par*

$$\frac{1}{p} + \frac{1}{q} = 1.$$

Dans la suite, p et q désigneront systématiquement deux exposants conjugués.

Exemple 1. *Le dual topologique des fonctions mesurables de puissance p intégrables L^p avec $p > 1$ est L^q.*

Les espaces L^p s'équipent naturellement de la fameuse « norme L_p » que nous mentionnons régulièrement et qui en fait un espace de Banach. Nous rappelons ses expressions usuelles ci-dessous.

Définition 2.2.3 (norme L_p). *La norme L_p est définie pour $f \in L_p(X, \mathcal{A}, \mu)$ par*

$$\|f\|_p = \left(\int_X |f|^p \, \mathrm{d}\mu \right)^{1/p}.$$

Si X est de dimension finie n on pose

$$\|x\|_p = \left(\sum_{i=1}^{n} |x^i|^p \right)^{1/p}.$$

À défaut de pouvoir identifier les éléments d'un espace X et de son dual X^*, on utilise le crochet de dualité pour désigner l'application d'un élément de X^* en un élément de X.

Définition 2.2.4 (Crochet de dualité). *Soit $x^* \in X^*$ et $x \in X$. On note $\langle x^*, x \rangle_{X^* \times X}$ ou $\langle x, x^* \rangle_{X \times X^*}$ le crochet de dualité défini par*

$$\langle x^*, x \rangle_{X^* \times X} = \langle x, x^* \rangle_{X \times X^*} = x^*(x).$$

Lorsque le contexte est clair, les indices indiquant les espaces dans lequel appartient chaque variable seront omis. Si l'espace est de dimension finie n, $\langle x^*, x \rangle_{X^* \times X}$ désignera le produit scalaire canonique $\langle x^*, x \rangle = \sum_{i=1}^{n} (x^*)^i x^i$

La classique inégalité de Cauchy-Schwarz reste vraie dans les espaces de Banach.

Théorème 2.2.5 (Inégalité de Cauchy)**.** *On a toujours*

$$|\langle x^*, x \rangle| \leq \|x^*\|_{X^*} \|x\|_X.$$

Notons cependant qu'elle résulte directement de la définition de la norme prise sur X^*. Dans les espaces L^p nous utiliserons l'inégalité plus générale de Hölder.

Théorème 2.2.6 (Inégalité de Hölder)**.** *Soit S un espace mesuré, $p, q > 0$ deux exposants conjugués, f un élément de $L^p(S)$ et g un élément de $L^q(S)$. Alors, le produit fg est un élément de $L^1(S)$ et on a l'inégalité*

$$\|fg\|_1 \leq \|f\|_p \|g\|_q.$$

L'opérateur de dualité permet de faire le lien entre les éléments de l'espace primal et ceux du dual. Il peut s'interpréter comme une généralisation du cas d'égalité de l'inégalité de Cauchy-Schwarz dans les espaces de Hilbert.

Définition 2.2.7 (Opérateur de dualité). *Soit $p \geq 1$. L'opérateur de dualité de X, à valeurs dans les parties de X^*, avec fonction de jauge $t \mapsto t^p$ est défini par*

$$J_p : X \to 2^{X^*}, x \mapsto \{x^* \in X^* : \langle x^*, x \rangle = \|x\| \|x^*\|, \|x^*\| = \|x\|^{p-1}\}$$

J_p est une fonction multivaluée en général. Il suffit que X soit un espace vectoriel normé pour que $J_p^X(x)$ soit non vide ([49]). Nous utiliserons l'abus de notation $J_p(x)$ pour désigner un élément de l'ensemble $J_p^X(x)$, tout en abandonnant l'exposant faisant référence à l'espace de travail quand le contexte le permet. Nous allons voir qu'en pratique J_p devient monovaluée sur les espaces de Banach courant (en particulier L^p pour $p > 1$).

Une propriété importante de l'opérateur de dualité J_p est qu'il correspond au sous différentiel d'une norme prise à l'exposant p [50] :

Théorème 2.2.8 (Théorème d'Asplund)**.** *Soit $p > 1$,*

$$J_p(x) = \partial(\frac{1}{p}\|\cdot\|^p).$$

En tant que sous-différentiel d'une fonction propre (finie en au moins un point), semi-continue inférieurement et convexe, J_p est un opérateur monotone ($\langle x^* - y^*, x - y\rangle \geq 0$ pour tout $x, y \in X$, $x^* \in J_p(x)$, $y^* \in J_p(y)$) maximal (il n'existe pas de fonction monotone qui contienne le graphe de J_p). On dispose d'une expression explicite lorsque $X = l^r$ ou $X = L^r$ avec $r > 1$:

Exemple 2.

$$\langle J_p^{l^r}(x), y\rangle = \sum_i \|x\|_r^{p-r}|x^i|^{r-1}\,\mathrm{signe}(x^i)y^i \tag{2.12}$$

$$\langle J_p^{L^r}(f), g\rangle = \int \|f\|_r^{p-r}|f(x)|^{r-1}\,\mathrm{signe}(f(x))g(x)\,dx. \tag{2.13}$$

En particulier si X est un espace de Hilbert, $J_2^X(x) = x$ et plus généralement $J_p^X(x) = \|x\|^{p-2}x$. L'expression (2.12) reste vraie pour $r = 1$ mais la fonction signe est modifiée pour devenir Signe, qui est essentiellement la même fonction sauf en 0 où $\mathrm{Signe}(0) = [-1; 1]$. Dans le cas de $(\mathbb{R}^n, \|\cdot\|_\infty)$ on a également

$$(J_p(x))^i = \|x\|_\infty^{p-1} z_i \tag{2.14}$$

avec $z_i = 0$ si $|x_i| \neq \|x\|_\infty (= \max_{1\leq i\leq N} |x_i|)$, $\mathrm{signe}(z_i) = \mathrm{signe}(x_i)$ et $\sum_i |z_i| = 1$. On peut prendre par exemple $y_i = \mathrm{signe}(x_i)$ pour un seul i tel que $|x_i| = \|x\|_\infty$ et $y_i = 0$ pour tous les autres i.

Afin d'être exhaustif, nous redonnons la liste des définitions des espaces (p-)strictement / uniformément convexes / lisses. En pratique, ce sont les liens établis aux Théorèmes (2.2.17) et (2.2.18) entre les propriétés de l'espace et celles de l'opérateur de dualité qui nous intéressera le plus.

Définition 2.2.9 (Espace p-convexe). *X est p convexe s'il existe une constante $c_p > 0$ telle que*

$$\frac{1}{p}\|x - y\|^p \geq \frac{1}{p}\|x\|^p - \langle J_p(x), y\rangle + \frac{c_p}{p}\|y\|^p, \text{ pour tout } x, y \in X.$$

Définition 2.2.10 (Espace p-lisse (p-smooth)). *X est p-lisse s'il existe une constante $G_p > 0$ telle que*

$$\frac{1}{p}\|x-y\|^p \leq \frac{1}{p}\|x\|^p - \langle J_p(x), y\rangle + \frac{G_p}{p}\|y\|^p, \text{ pour tout } x, y \in X.$$

Définition 2.2.11 (Espace strictement convexe). *X est strictement convexe si $\|\frac{1}{2}(x+y)\| < 1$ pour tout x, y de la sphère unité de X avec $x \neq y$*

Définition 2.2.12 (Espace uniformément convexe). *X est uniformément convexe si, pour le module de convexité $\delta_X : [0;2] \to [0;1]$ défini par*

$$\delta_X(\varepsilon) = \inf\{1 - \|\frac{1}{2}(x+y)\| : \|x\| = \|y\| = 1,\ \|x-y\| \geq \varepsilon\}$$

on a

$$\forall 0 < \varepsilon \leq 2,\ \delta_X(\varepsilon) > 0$$

Définition 2.2.13 (Espace lisse/smooth). *X est lisse si pour chaque $x \in X$, $x \neq 0$, il existe une unique forme linéaire $x^* \in X^*$ telle que $\|x^*\| = 1$ et $\langle x^*, x\rangle = \|x\|$.*

Exemple 3. *On sait que ([51]) l^p et L^p, équipés de leur norme usuelle et avec $1 < p < \infty$, sont*

— $\max(2,p)$*-convexes*

et

— $\min(2,p)$*-lisses.*

Définition 2.2.14 (Espace uniformément lisse). *X est uniformément lisse si, pour le module de régularité (modulus of smoothness) $\rho : [0;\infty[\to [0;\infty[$ défini par*

$$\rho_X(\tau) = \frac{1}{2}\sup\{\|x+y\| + \|x-y\| - 2 : \|x\| = 1, \|y\| \leq \tau\}$$

on a la limite

$$\lim_{\tau \to 0^+} \frac{\rho_X(\tau)}{\tau} = 0.$$

Exemple 4. *Les espaces L_p ($1 < p < +\infty$) sont uniformément convexes et uniformément lisses et*

$$\rho_{L_p}(\tau) = \begin{cases} (1+\tau^p)^{\frac{1}{p}} - 1, & 1 < p \leq 2 \\ \frac{p-1}{2}\tau^2 + o(\tau^2), & p > 2 \end{cases}. \tag{2.15}$$

Si un espace de Banach est p-lisse avec $p > 1$, la norme à la puissance p est différentiable au sens de Fréchet, donc également au sens de Gâteaux, et $J_p(x)$ sera un singleton pour tout x.

L'espace l^1 n'est ni p-convexe, ni p-lisse et ce quel que soit p. En revanche les espaces de Hilbert l^2 et L^2 sont 2-convexes et 2-lisses d'après l'identité de polarisation $\frac{1}{2}\|x-y\|_2^2 = \frac{1}{2}\|x\|_2^2 - \langle x, y\rangle + \frac{1}{2}\|y\|_2^2$.

Il faut noter que ces différentes définitions ont un lien entre elles ([52]) :

Théorème 2.2.15. *Si X est p-convexe alors les propositions suivantes sont équivalentes*

- *$p \geq 2$,*
- *X est strictement et uniformément convexe,*
- *X est réflexif.*

Si X est p-lisse alors les propositions suivantes sont équivalentes

- *$p \leq 2$,*
- *X est strictement et uniformément lisse,*
- *X est réflexif.*

De même les propriétés vérifiées par X sont en lien avec celles vérifiées par X^* :

Théorème 2.2.16.
- *X est p-lisse si et seulement si X^* est q-convexe.*
- *X est p-convexe si et seulement si X^* est q-lisse.*
- *X est uniformément convexe (respectivement uniformément lisse) si et seulement si X^* est uniformément lisse (respectivement uniformément convexe).*

Nous pouvons finalement énoncer les théorèmes qui nous seront les plus utiles. les propriétés de l'espace renseignent sur les propriétés vérifiées par l'opérateur de dualité (et réciproquement) :

Théorème 2.2.17.
1. *Pour tout $x \in X$, $J_p(x)$ est non vide et convexe.*
2. *$J_p(-x) = -J_p(x)$ et $J_p(\lambda x) = \lambda^{p-1} J_p(x)$ pour tout $x \in X$ et pour tout $\lambda > 0$.*
3. *Pour tout $p, q > 1$ on a $\|x\|^{q-1} J_p(x) = \|x\|^{p-1} J_q(x)$*
4. *Si X est p-convexe et lisse, J_p est monovaluée et bijective, $J_q^{X^*}(x)$ est monovaluée et*

$$J_q^{X^*}(J_p(x)) = x.$$

5. *Soit M un espace fermé et convexe de X. Si X est uniformément convexe alors il existe un unique $x \in M$ tel que*

$$\|x\| = \inf_{z \in M} \|z\|.$$

La quatrième propriété de ce dernier théorème est fondamentale pour transporter les itérés du primal au dual. Il explique également pourquoi les algorithmes que nous allons voir se placent dans le cadre $p > 1$. La proposition suivante, parfois désignée comme l'une des inégalités de Xu-Roach, a un corollaire essentiel.

Théorème 2.2.18. *Les propositions suivantes sont équivalentes :*

1. *X est s-lisse.*
2. *Pour un certain $p \in]1;+\infty[$, J_p^X est monovalué et pour tout $x, y \in X$ on a*

$$\|J_p^X(x) - J_p^X(y)\| \leq C \max(\|x\|, \|y\|)^{p-s} \|x - y\|^{s-1} \tag{2.16}$$

   avec $C > 0$ une constante indépendante de x et y.
3. *La proposition 2. est vraie pour tout $p \in]1;+\infty]$*

Son corollaire que nous utiliserons s'énonce

Corollaire 2.2.19. *Soit X un espace s-lisse et $p > 1$. Alors l'opérateur de dualité J_p^X est monovalué et $\min(p-1, s-1)$-Hölder continu sur les ensembles bornés, c'est à dire qu'il existe $C, \tilde{C} > 0$ tels que*

$$\|J_p^X(x) - J_p^X(y)\| \leq C \max(\|x\|, \|y\|)^{p-s} \|x - y\|^{s-1} \leq \tilde{C} \|x - y\|^{s-1}. \tag{2.17}$$

Enfin un outil souvent utilisé car plus maniable que la distance en norme pour l'analyse de convergence dans les espaces de Banach est la distance de Bregman :

Définition 2.2.20 (Distance de Bregman). *Pour tout $x, y \in X$ la distance de Bregman est définie par*

$$\Delta_p(x, y) = \frac{1}{q} \|J_p(x)\|^q - \langle J_p(x), y \rangle + \frac{1}{p} \|y\|^p.$$

Elle est utilisée dans les algorithmes présentés à ce chapitre. Il ne s'agit pas d'une distance en général, néanmoins elle a les propriétés suivantes :

Théorème 2.2.21.
1. *Pour tout $x, y \in X$, $\Delta(x, y) \geq 0$.*
2. *Pour tout $p > 1$ et pour toute suite $(x_n)_n$ à valeur dans X, $\lim_{n\to\infty} \Delta_p(x_n, x) = 0 \Leftrightarrow \lim_{n\to\infty} \|x_n - x\| = 0$.*

Exemple 5. *Pour un espace de Hilbert $\Delta_2(x, y) = \frac{1}{2}\|x - y\|^2$.*

Investiguons à présent les différents algorithmes de minimisation dans les espaces de Banach qui ont été proposés.

2.2.2 Descente de gradient dans les espaces de Banach

Schöpfer *et al.* [70] s'intéressent à la résolution d'un système $Ax = y$ avec $A : X \to Y$ un opérateur linéaire continu et $y \in Im(A)$. Notons, par analogie avec le cas où $X = L^p$ et $Y = L^r$, $J_p : X \to X^*$ l'opérateur de dualité transportant les itérés dans l'espace dual de X et $J_q : X^* \to X$ l'opérateur inverse ramenant les itérés dans l'espace primal (et de même pour $J_r : Y \to Y^*$). Le schéma itératif proposé repose sur le transport des itérés entre le primal X et le dual X^* :

$$x_0^* = J_p(x_0) \tag{2.18}$$
$$x_{k+1}^* = x_k^* - \alpha_k A^* J_r(Ax_n - y) \tag{2.19}$$
$$x_{k+1} = J_q(x_{k+1}^*). \tag{2.20}$$

Les itérés appartenant à X^* sont notées avec une étoile en exposant. Ces itérations généralisent celles de Landweber pour des espaces de Banach. Le même article traite le cas de données exactes et de données approximatives $(y_k)_k$ et $(A_l)_l$. Cette dernière hypothèse permet par exemple de prendre en compte l'erreur modèle d'un système d'assimilation. Les auteurs font l'hypothèse que les erreurs peuvent être estimées :

$$\|y_k - y\| \leq \delta_k, \qquad \delta_k > \delta_{k+1} > 0, \qquad \lim_{k\to\infty} \delta_k = 0$$
$$\|A_l - A\| \leq \eta_l, \qquad \eta_l > \eta_{l+1} > 0, \qquad \lim_{l\to\infty} \eta_l = 0.$$

On suppose également que l'on a un majorant de la norme de la solution :

$$\|\bar{x}\| \leq R$$

et on pose

$$S = \sup_{l\in\mathbb{N}} \|A_l\|.$$

L'algorithme s'écrit alors

Algorithme 4 Algorithme de Schöpfer *et al.*

1: Fixer $p, r \in]1; +\infty$ et choisir deux constantes $C, D \in]0; 1[$
2: Prendre un vecteur initial $x_0 \in X$ tel que $J_p(x_0) \in \overline{Im(A*)}$ et $\Delta_p(x_0, \bar{x}) \leq \frac{1}{p}\|x\|^p$. On pose $k_{-1} = l_{-1} = 0$.
3: $k = 0$
4: **si** Pour tout $k > k_{n-1}$ et $l > l_{n-1}$ on a $\|A_l x_n - y_k\| \geq \frac{1}{D}(\delta_k + \eta_l R)$ **alors**
5: Stop
6: **sinon**
7: On peut trouver $k_n > k_{n-1}$ et $l_n > l_{n-1}$ avec $\delta_{k_n} + \eta_{l_n} R \leq DR_n$ où $R_n = \|A_{l_n} x_n - y_{k_n}\|$. On pose alors :
8: Si $x_0 = 0, \mu_0 = C(1-D)^{p-1} \frac{q^{p-1}}{S^p} R_0^{p-r}$
9: Pour tout $n \geq 0$ (resp. $n \geq 1$ si $x_0 = 0$), on pose $\lambda_n = \max(\rho_{X^*}(1), \frac{C(1-D)}{2^q G_q S} \frac{R_n}{\|x_n\|})$
10: Choisir $\tau_n \in]0; 1]$ avec $\frac{\rho_{X^*}(\tau_n)}{\tau_n} = \lambda_n$
11: Poser $\mu_n = \frac{\tau_n}{S} \frac{\|x_n\|^{p-1}}{R_n^{r-1}}$
12: Mettre à jour $J_p(x_{n+1}) = J_p(x_n) - \mu_n A_{l_n}^* J_r(A_{l_n} x_n - y_{k_n})$,
13: $x_{n+1} = J_q(J_p(x_{n+1}))$
14: **fin si**

En faisant seulement l'hypothèse que X est uniformément convexe, les auteurs obtiennent la convergence forte des itérations vers l'unique solution de norme minimale du problème.

Un an plus tard Boneski *et al.* [71] cherchent à calculer une solution de $Ax = y + \eta$ (ou η représente un bruitage) sous les mêmes hypothèses. Pour ce faire, ils s'intéressent à la minimisation d'une fonctionnelle pénalisée $\Psi : X \to \mathbb{R}$ donnée par

$$\Psi(x) = \frac{1}{r}\|Ax - y\|_Y^r + \frac{\lambda}{p}\|x\|_X^p. \tag{2.21}$$

Deux schéma itératifs sont considérés : le premier, similaire à l'algorithme de Schöpfer *et al.*, va transporter les itérés dans l'espace dual :

$$x_{k+1}^* = x_k^* - \alpha_k \psi_k, \text{ avec } \psi_k \in \partial\Psi(x_k) \tag{2.22}$$
$$x_{k+1} = J_q(x_{k+1}^*). \tag{2.23}$$

Le second transporte la direction de descente dans l'espace primal :

$$x_{k+1} = x_k - \alpha_k J_q^*(\nabla\Psi(x_k)). \tag{2.24}$$

Pour le premier schéma (2.22), le pas est tel que

$$\alpha_k = \frac{\lambda}{pGP}\varepsilon \tag{2.25}$$

où $P = \sup\{\|\psi_k\|^2 : \psi \in \partial\Psi(x) \text{ avec } \Delta_2(\bar{x}, x) \leq R\}$ (Δ est la distance de Bregman) avec $\bar{x}$ le minimiseur de Ψ, R tel que $\Delta_2(x_0, \bar{x}) \leq R$ et G_p une constante vérifiant la définition (2.2.10) des espace p-lisses (sa valeur exacte peut être trouvée dans [52]). Quant à ε, il est pris égal à

$$\varepsilon = \frac{-1 + \sqrt{1 + 4D\varepsilon_{aim}}}{2D} \tag{2.26}$$

avec ε_{aim} la précision recherchée entre $\bar{x}$ et l'itéré final (au sens de la distance de Bregman) et $D = \frac{\lambda^2}{2p^2GP}$. Ce schéma requiert donc l'évaluation de P (étape non triviale en général) et ne certifie pas une décroissance monotone de $(\Psi(x_k))_k$, mais converge bien fortement vers l'unique solution du problème, de même que le second schéma 2.24.

Ce second schéma suppose quant à lui une recherche linéaire du pas le long de la direction $-\nabla\Psi_k$ ramenée dans le primal, i.e. le calcul de

$$\Psi(x_k - \alpha_k J_q^*(\nabla\Psi(x_k)) = \min_{\alpha\in\mathbb{R}} \Psi(x_k - \alpha J_q^*(\nabla\Psi_k)).$$

Mais une recherche linéaire produisant un pas vérifiant

$$f_{k+1} \leq c\alpha_k \|\nabla f_k\|^q$$

avec $c \in]0; 1[$ suffit pour la convergence.

De même que l'algorithme du gradient conjugué est une amélioration de la descente de gradient classique dans le sens où il apporte plus d'information pour la construction de la direction de descente à chaque étape, en conséquence de quoi il est généralement plus efficace, de même il existe des algorithmes de gradient conjugués dans les espaces de Banach.

2.2.3 Algorithmes de gradient conjugué dans les espaces de Banach

Algorithme d'Herzog et Wollner

Nous revenons au problème de la résolution de $Ax = y$ avec $A : X \to Y$, X et Y deux espaces de Banach et $y \in Im(A)$. Un algorithme de gradient conjugué dans un espace de Banach est un algorithme dont la direction de descente p_k à l'itération k va dépendre des directions de descente précédentes. On ne peut pas conserver toutes les propriétés du gradient conjugué classique : pour conserver celle de A conjugaison des directions de descente il est nécessaire de faire dépendre p_k de toutes les directions précédentes. C'est la méthode proposée par Herzog et Wollner [72].

Algorithme 5 Algorithme du gradient conjugué d'Herzog et Wollner

$r_0 = Ax_0 - b$
$p_0 = -J_p(r_0)$
$k = 0$
tant que non convergence **faire**
  $\alpha_k = \frac{\langle r_k, p_k \rangle}{\langle Ap_k, p_k \rangle}$
  $x_{k+1} = x_k + \alpha_k p_k$
  $r_{k+1} = r_k + \alpha_k A p_k$
  $\beta_{k+1,i} = \frac{\langle Ap_i, J_p(r_{k+1}) \rangle}{\langle Ap_i, p_i \rangle}$ for $i = 0, ..., k$
  $p_{k+1} = J_p(r_{k+1}) + \sum_{i=0}^{k} \beta_{k+1,i} p_i$
  $k = k + 1$
fin tant que

Cet algorithme partage de nombreuses autres propriétés avec l'algorithme du gradient conjugué classique dans les espaces de Hilbert : les directions sont A conjuguées ($\langle Ap_j, p_k \rangle = 0$ pour $0 \leq j < k$), le résidu est « orthogonal » (au sens du crochet de dualité $\langle \cdot, \cdot \rangle$ qui ne désigne plus forcément le produit scalaire euclidien) aux directions de descente ($\langle r_{k+1}, p_j \rangle = 0$ pour $0 \leq j < k$), le pas α_k est l'exact minimiseur de la quadratique uniformément convexe $\phi(x) = \frac{1}{2}\langle Ax, x \rangle - \langle b, x \rangle$ et x_{k+1} vérifie $x_{k+1} = \underset{p \in \text{vect}\{p_0, ... p_k\}}{\arg\min} \phi(x_0 + p)$.

Cette approche peut être néanmoins coûteuse en espace mémoire puisqu'elle nécessite le stockage de toutes les directions de descentes calculées. Dans le même article, les auteurs proposent une version à mémoire limitée de l'algorithme qui n'entrave pas la

vitesse de convergence pour leur application lorsque les 3 dernières directions de descente sont retenues à chaque itération.

Algorithme d'Estatico *et al.*

Une autre démarche, que l'on appellera l'algorithme du gradient conjugué d'Estatico, proposée par Estatico *et al.* dans [73] garde la dépendance entre p_{k+1} et p_k uniquement. Toujours dans le cas d'un espace X uniformément convexe le problème admet une unique solution de norme p minimale (grâce à l'assertion 5. du Théorème 2.2.17), vers laquelle va converger fortement les itérés issus de l'algorithme proposé dans le cas où $X = L^p$ et $Y = L^r$:

Algorithme 6 Algorithme du gradient conjugué d'Estatico *et al.*

Choisir $C \in (0,1)$ et poser $\gamma = \frac{Cr}{2^r - 1 + Cr}$ tel que
$0 < d \leq (1 - \frac{2^r - 1 + r}{r}\gamma)\frac{1}{\|A\|}$
Poser $k = 0$, $x_0^* = 0$, $p_0 = A^* J_r(b)$, $R_0 = \|b\|$ et
$\alpha_0 = \underset{\alpha \in [0, \frac{q^{p-1}}{\|A\|^p R_0^{p-r}}]}{\arg\min} \|AJ_q(x_0^* + \alpha p_0) - b\|^r$
tant que $R_k > 0$ **faire**
    $k = k + 1$
    $x_k^* = x_{k-1}^* + \alpha_{k-1} p_{k-1}^*$
    $x_k = J_q(x_k^*)$
    $p_k = -A^* J_r(Ax_k - b) + \beta_k p_{k-1}$
    où $\alpha_k = \underset{\alpha \in [0, T_k]}{\arg\min} \|AJ_q(x_k^* + \alpha p_k) - b\|^r$
    et $\beta_k = \gamma \frac{R_k^r}{R_{k-1}^r}$
    avec $R_k = \|Ax_k - b\|$
    $T_k = \min\{\frac{R_k^{2-r} = (V_k - d\|A\|Q_k)}{G_q 2^{q-2} \|x_k^*\|^{q-2} (q-1) \|A\|^2 Q_k^2}, \frac{\|x_k^*\|}{\|A\| R_k^{r-1} Q_k}\}$
    $V_k = 1 - \frac{2^r - 1}{r}\gamma Q_{k-1}$
    $Q_k = \frac{1 - \gamma^{k+1}}{1 - \gamma}$
    et G_q la constante de l'espace de Banach L_q définie dans [52]
fin tant que

Cette fois les directions n'ont plus de raison d'être A-conjuguées. La preuve de la convergence repose néanmoins sur l'appartenance du pas α à un intervalle très particulier. Plusieurs constantes doivent être également choisies et leur impact sur le comportement de l'algorithme n'a pas été approfondi.

L'algorithme s'adapte à des données bruitées b_δ telles que $\|b - b_\delta\| \leq \delta$ avec un bruit $\delta > 0$ connu, en modifiant les valeurs de γ, d et V_k et la condition d'arrêt. Cependant

même avec des données exactes, l'algorithme impose une forme de régularisation implicite au cours des itérations car celui-ci converge vers la solution de norme p minimale.

Les étapes nécessitant l'évaluation de $\|\mathbf{A}\|$ en tant que norme d'opérateur peuvent s'effectuer via un algorithme de puissance itéré. Boyd [74] propose un algorithme convergeant vers $\|A\|_{r,p} = \max\{\|Ax\|_p : \|x\|_r = 1\}$. Dans le cas où $p = r$, Higham [75] fournit un algorithme donnant une estimation à $n^{1-\frac{1}{p}}$ de la norme $\|A\|_p$. Ces algorithmes nécessitent l'évaluation du produit matrice-vecteur $\mathbf{Ax}$ et fournissent en pratique une estimation acceptable en quelques itérations.

2.3 Conclusion

Nous avons vu quels algorithmes étaient actuellement utilisés dans les centres d'assimilation de données. Ils se divisent en deux grandes classes : ceux qui minimisent directement le 4DVar et ceux qui font appel à deux boucles imbriquées : une externe de linéarisation et une interne de minimisation du 4DVar linéarisé. Pour les schémas avec boucles externes et internes, la régularisation s'effectue dans la boucle interne.

Nous avons également présenté un formalisme inusité jusqu'à maintenant dans ce domaine pour minimiser des fonctionnelles dans des espaces de Banach. Son intérêt est précisé au chapitre 4, où, dans le même temps, nous exploiterons les concepts introduits ici pour proposer de nouveaux algorithmes et montrer leur convergence. Ce formalisme est bien sûr en lien avec la pénalisation en norme L_p. Il s'agit donc avant tout de voir pourquoi et quand utiliser une telle régularisation en assimilation de données.

Deuxième partie

Contributions : utilisation de la norme L_p en assimilation de données et études de nouveaux algorithmes de minimisation adaptés à cette régularisation

Chapitre 3

La régularisation en norme L_p en assimilation de données

3.1 Pourquoi régulariser avec une norme L_p 61
3.1.1 Intérêt de modélisation statistique 61
3.1.2 Intérêts numériques de la norme L_p 65
3.2 Illustration sur un problème d'assimilation de données : l'advection 1D . . 65
3.2.1 Contexte expérimental 65
3.2.2 Choix des paramètres λ et p 69
3.2.3 Résultats pour le scénario parfait 71
3.2.4 Résultats pour le modèle imparfait 73
3.2.5 Bilan de l'expérience d'advection 79
3.3 Conclusion 83

Nous allons maintenant justifier l'utilisation d'une norme L_p avec $1 < p < 2$ en assimilation de données, d'un point de vue théorique et aussi expérimental à travers un modèle-jouet : l'équation d'advection en une dimension. Les résultats de ce chapitre sont majoritairement issus de Bernigaud *et al.* [76].

3.1 Pourquoi régulariser avec une norme L_p

3.1.1 Intérêt de modélisation statistique

Le cas de la modélisation de la dérivée de l'épaisseur de la glace

Dans un article de 2019 [77], les auteurs observent que la dérivée de l'épaisseur de la glace dans la mer de Beaufort est plus susceptible de suivre une loi gaussienne généralisée (abrégée GGD pour Generalized Gaussian Distribution) qu'une loi normale standard. Sa densité est donnée par (voir par exemple [78]) :

$$f(x;\alpha,p) = \frac{p}{2\alpha\Gamma(1/p)}\, e^{-\left(\frac{|x-\mu|}{\alpha}\right)^p}. \tag{3.1}$$

Le paramètre α joue le rôle d'un facteur d'échelle tandis que p joue le rôle d'un paramètre de forme. $\Gamma(\cdot)$ est la fonction Gamma. Nous avons vu à la section (1.2.1) que la régularisation en norme L_2 était associée à des hypothèses d'erreurs suivant des lois gaussiennes. Naturellement, la norme L_p sera associée à des hypothèses d'erreurs suivant des lois gaussiennes généralisées. Pour continuer nous devons définir ce qu'est une loi gaussienne en dimension supérieure à un. La définition la plus intuitive est donnée par Goodman ([79]) et s'apparente à celle d'un vecteur gaussien généralisé, en définissant d'abord la loi gaussienne généralisée multivariée standard.

Définition 3.1.1 (Loi gaussienne généralisée multivariée standard). *Une loi gaussienne généralisée standard*

$$\mathbf{x} = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

de taille $n \times 1$ consiste en n variables aléatoires $(x_i)_{1\leq i\leq n}$ suivant des lois (marginales) gaussiennes généralisées indépendantes et identiquement distribuées.

Définition 3.1.2 (Loi gaussienne généralisée multivariée). *Un vecteur $\mathbf{y}$ de taille $n \times 1$ suit alors une loi gaussienne généralisée $\mathcal{GGD}(\boldsymbol{\mu}, \mathbf{C}, p)$ si et seulement si, par définition, $\mathbf{y}$ s'écrit*

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \boldsymbol{\mu}.$$

avec $\boldsymbol{\mu} \in \mathbb{R}^n$,$p > 0$ et $\mathbf{C} \in GL_n(\mathbb{R})$ des paramètres fixés et $\mathbf{x}$ un vecteur suivant une loi gaussienne généralisée multivariée standard.

La densité de probabilité d'une variable aléatoire $\mathbf{y}$ de taille $n \times 1$ suivant une loi gaussienne généralisée multivariée $\mathcal{GGD}(\boldsymbol{\mu}, \mathbf{C}, p)$ est alors donnée par

$$f(\mathbf{y}) = K(n, \mathbf{C}, p) \exp\left(-\sum_{i=1}^{n} |\sum_{j=1}^{n} (\mathbf{C})^{-1}_{ij} (\mathbf{y}_i - \boldsymbol{\mu}_j)|^p \right), \tag{3.2}$$

où $K(n, \mathbf{C}, p) = \left[(2\Gamma(1+\frac{1}{p}))^n \det(\mathbf{C}) \right]^{-1}$.

Remarque 1. *Avec cette définition, on peut montrer que $\mathcal{GGD}(\boldsymbol{\mu}, \mathbf{C}, 2) = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ la loi normale de moyenne $\boldsymbol{\mu}$ et de matrice de variance-covariance Σ telle que $\Sigma = \frac{1}{2}\mathbf{C}\mathbf{C}^T$.*

Il existe une autre manière de définir une loi gaussienne généralisée multivariée, proposée par De Simoni ([80]), en reprenant cette fois directement la densité de probabilité d'une loi normale, mais en portant la puissance de l'exponentielle à p au lieu de 2.

Définition 3.1.3 (Loi gaussienne généralisée multivariée - définition 2). *La densité de probabilité d'une telle loi s'écrit alors pour tout* $\mathbf{x}$ *dans* $\mathbb{R}^n$ *:*

$$p(\mathbf{x}|\mathbf{M}, m, p) = \frac{1}{det(\mathbf{M})} h_{m,p}(\mathbf{x}^T \mathbf{M}^{-1} \mathbf{x}) \tag{3.3}$$

avec $\mathbf{M}$ *une matrice symétrique* $n \times n$ *symétrique,* m *et* p *les paramètres d'échelle et de forme respectivement et* $h_{m,p}$ *la fonction définie par, pour tout* y *dans* $\mathbb{R}$*,*

$$h_{m,p}(y) = \frac{p\Gamma(\frac{n}{2})}{\pi^{\frac{n}{2}}\Gamma(\frac{n}{2p})2^{\frac{n}{2p}}} \frac{1}{m^{\frac{n}{2}}} \exp\left(-\frac{y^p}{2m^p}\right) \mathbb{1}_{\mathbb{R}^+}(y)$$

où $\mathbb{1}_{\mathbb{R}^+}$ *est la fonction indicatrice de* $\mathbb{R}^+$*.*

Pour exploiter les observations apportées dans [77] sur la mer de Beaufort, modélisons l'épaisseur de la glace par la variable d'état $\boldsymbol{X}$ et supposons comme d'habitude que $\boldsymbol{Y} = \mathbf{H}\boldsymbol{X} + \boldsymbol{N}$ avec $\boldsymbol{X} \sim \mathcal{N}(\mathbf{x}_b^0, \mathbf{B})$ et $\boldsymbol{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$. Pour ajouter la nouvelle information a priori concernant la dérivée de l'épaisseur de la glace, posons pour $i = 0, ..., n-1$

$$x_{i+1} = x_i + \varepsilon_i^p$$

avec $\epsilon_i^p \sim \mathcal{GGD}(\mu = 0, \alpha, p)$, ou encore $p(z_i \equiv x_{i+1} - x_i) = \frac{p}{2\alpha\Gamma(1/p)} e^{-(\frac{|z_i|}{\alpha})^p}$ et α un paramètre d'échelle. Posons toujours

$$\mathbf{\Phi} = \begin{bmatrix} 1 & & & 0 \\ -1 & \ddots & & \\ & \ddots & \ddots & \\ 0 & & -1 & 1 \end{bmatrix} \tag{3.4}$$

la matrice de dérivation numérique. Sous forme matricielle nous avons donc $\boldsymbol{Z} \equiv \mathbf{\Phi} X \sim \mathcal{GGD}(\mathbf{0}, \alpha\mathbf{I}, \mathbf{p})$.

Supposons que les erreurs considérées soient mutuellement indépendantes. La loi de Bayes donne alors

$$p_{\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{z} \mid \boldsymbol{y}) \propto p_{\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{Z}}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{z}) p_{\boldsymbol{Z}|\boldsymbol{X}}(\boldsymbol{z} \mid \boldsymbol{x}) p_{\boldsymbol{X}}(\boldsymbol{x}).$$

Étant en mesure de déduire $\boldsymbol{Y}$ directement à partir de $\boldsymbol{X}$, on a $p_{\boldsymbol{Y}|\boldsymbol{X},\mathbf{\Phi}\boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{x}, \mathbf{\Phi}\boldsymbol{x}) = p_{\boldsymbol{Y}|\boldsymbol{X}}(\boldsymbol{y} \mid \boldsymbol{x})$. Le maximum a posteriori va donc minimiser l'opposé de la log-vraisemblance, soit

$$\min\{\frac{1}{2} \parallel \mathbf{y} - \mathbf{H}\mathbf{x} \parallel^2_{\mathbf{R}^{-1}} + \frac{1}{2} \parallel \mathbf{x} - \mathbf{x}^0_b \parallel^2_{\mathbf{B}^{-1}} + \frac{1}{\alpha^p} \parallel \mathbf{\Phi}\mathbf{x} \parallel^p_p\}. \tag{3.5}$$

Supposer que la dérivée (numérique) de l'épaisseur de la glace $\mathbf{\Phi}X$ suit une loi normale généralisée est donc équivalent à minimiser la distance entre deux composantes consécutives de X dans une certaine norme. Il est donc pertinent dans ce cas de figure de rajouter une pénalisation en L_p sur la dérivée du signal.

Remarque 2. *Pour estimer les paramètres α et p d'une loi gaussienne généralisée (3.1), on peut se référencer à [81]. Des estimateurs de paramètres ont également été proposés dans le cas d'une loi normale généralisée multivariée pour les deux définitions mentionnées : dans [79] pour la première définition et dans [82] pour la seconde.*

Autre point de vue : régularisation par un terme d'écart à l'ébauche en norme L_p

Toujours dans le cadre de la recherche du maximum a posteriori de la section (1.2.1), on voit qu'on aurait également pu imposer $X \sim \mathcal{GGD}(\mathbf{x}^b, \mathbf{C}, p)$ au lieu de $X \sim \mathcal{N}(\mathbf{x}^b, \frac{1}{2}\mathbf{C}\mathbf{C}^T)$ dans le cas où la variable X suivrait une loi gaussienne généralisée multivariée au sens de la première définition (3.1.2). Cela mène directement à minimiser la somme d'un terme d'écart aux observations classique et d'un terme d'écart à l'ébauche en norme L_p :

$$\Omega(\mathbf{x}_0) = \|\mathbf{x}_0 - \mathbf{x}^b\|^p_{p,\mathbf{C}^{-2}} + \frac{1}{2}\|\mathbf{y} - \hat{\mathcal{H}}(\mathbf{x}_0)\|^2_{\mathbf{R}^{-1}}. \tag{3.6}$$

Le terme $\|\mathbf{x}_0 - \mathbf{x}^b\|^p_{p,\mathbf{C}^{-2}}$ correspond à moins la log-vraisemblance de (3.2), c'est à dire $\|\mathbf{C}^{-1}(\mathbf{x}_0 - \mathbf{x}^b)\|^p_p$. Il est encore possible à ce stade de rajouter en plus un terme de pénalisation sur la variable d'état (en n'importe quelle norme). Nonobstant une pénalisation supplémentaire, par la présence d'un terme de régularisation en norme L_p, les algorithmes que nous présenterons au chapitre (4) gardent tout leur intérêt dans ce cas de figure.

Comme cadre d'application on peut faire référence à l'article de Banerjee *et al.* ([83]) dans lequel les auteurs montrent que le bruit acoustique sous-marin était mieux représenté par une loi gaussienne généralisée qu'une loi normale.

Qu'advient-il si nous considérons la seconde définition de la loi normale multivariée (3.1.3) ? L'opposé de la log-vraisemblance de la densité (3.3) amène à minimiser, à un facteur multiplicatif près, $\|\mathbf{x}_0 - \mathbf{x}^b\|^p_{\mathbf{B}^{-1}} = \|\mathbf{B}^{-\frac{1}{2}}(\mathbf{x}_0 - \mathbf{x}^b)\|^p_2$. C'est maintenant la norme L_2 qui intervient via le terme $\mathbf{x}^T\mathbf{M}\mathbf{x}$ de la définition, élevée à la puissance p. Nous n'approfondissons cependant pas cette formulation par la suite.

3.1.2 Intérêts numériques de la norme L_p

Même sans justification statistique, un terme de régularisation en norme L_p est toujours envisageable. Nous avons déjà vu au premier chapitre (Section 1.3.3) quelques raisons pour son utilisation : prendre p proche de 1 permet notamment de réduire les oscillations qu'une norme L_1 introduirait dans la reconstruction d'un signal discontinu, tout en conservant une forte pénalisation sur les composantes proches de zéros.

Plus généralement elle réalise un compromis entre les normes L_1 et L_2. Alors que ces deux normes sont appropriées pour des signaux respectivement discontinus et lisses (par exemple un signal carré et un signal sinusoïdal), la norme L_p peut être considérée pour des signaux dont la régularité est intermédiaire (un signal trapézoïdal, un signal triangulaire ...). Ces signaux « quasi-parcimonieux » (que nous dénommons également « quasi-creux ») sont présents en assimilation de données aux travers des exemples déjà donnés dans la section précédente, mais également au travers de la concentration de la sargasse dans la mer qui présente cette structure trapézoïdale ([84]), ou encore de fronts météorologiques ([33]). L'objet de la prochaine section est donc l'illustration et l'étude de tels intérêts.

3.2 Illustration sur un problème d'assimilation de données : l'advection 1D

Ce problème simple qu'est celui de l'advection en une dimension va nous permettre de mettre en évidence les bénéfices d'utiliser une telle régularisation. Il est également intéressant car il nous permet de simuler facilement des phénomènes de lissage du signal au cours du temps que l'on retrouve sur des systèmes physiques plus élaborés (équations de diffusion, de Navier-Stokes ...). Il sert donc dans un premier temps de proxy à un vrai système d'assimilation : nous espérons que les propriétés que nous allons mettre en avant sur ce modèle jouet se propageront lorsque la dimension et la complexité du système augmenteront (ce qui sera étudié au chapitre 5).

3.2.1 Contexte expérimental

Les performances de la régularisation en norme L_p sont évaluées pour différentes valeurs de p sur des expériences jumelles portant sur l'équation d'advection linéaire 1D :

$$\begin{cases} \partial_t u(s,t) + c\partial_s u(s,t) = 0 \\ u(s,t_0) = u_0(s). \end{cases} \tag{3.7}$$

Les conditions aux frontières sont fixées par :

$$u(0,t) = u(L,t) = 0 \tag{3.8}$$

avec une longueur d'intervalle $L = 1$, mais l'intervalle de temps considéré est pris suffisamment petit pour que le support du signal, soumis à l'équation d'advection, n'atteigne

pas les bords de la grille lors de la simulation. En effet, sans effet de bord, cette équation possède la solution analytique $u(s,t) = u_0(s-ct)$: la solution au temps t est simplement la condition initiale translatée par la distance ct.

L'équation est discrétisée grâce à un schéma de Lax-Wendroff qui est d'ordre deux en temps et en espace ([85]) :

$$\begin{aligned} u_{i+1}(j) &= u_i(j) - \frac{\mu}{2}\left[u_i(j+1) - u_i(j-1)\right] \\ &\quad + \frac{\mu^2}{2}\left[u_i(j+1) - 2u_i(j) + u_i(j-1)\right] \end{aligned} \tag{3.9}$$

avec $\mu = \dfrac{c\Delta t}{\Delta s}$ le nombre de Courant et i (respectivement j) l'indice de numérotation temporel (respectivement spatial). Les valeurs des paramètres sont listés dans la Table 3.1. Deux cas sont considérés selon la valeur de Δt : un scénario « modèle parfait » qui correspond à un nombre de Courant $\mu = 1$ et un scénario « modèle imparfait » qui correspond à un nombre de Courant $\mu = 0.5$.

En effet, de la diffusion implicite est introduite par le modèle numérique lorsque le nombre de Courant est strictement inférieur à 1. Dans le scénario du modèle imparfait, le modèle numérique va alors lisser le signal au cours du temps et simule une erreur modèle commise lors de l'assimilation. Deux sous-cas seront également considérés : celui d'un signal initial vrai parcimonieux et celui d'un signal initial vrai quasi-parcimonieux. Dans ce cadre, nous nous attendons à ce que l'effet d'une pénalisation en norme L_1 sur l'état analysé soit mitigé lors de la seconde partie de la fenêtre d'assimilation, durant laquelle le signal aura perdu sa parcimonie initiale. Au contraire, lors du scénario parfait le modèle numérique va maintenir la parcimonie du signal au cours du temps et le signal ne sera pas lissé par le modèle au cours du temps.

	modèle imparfait	modèle parfait
Δs	0.01	0.01
Δt	0.005	0.01
c	1	1
L	1	1
μ	0.5	1

TABLE 3.1 – Paramètres de discrétisation

Des mesures tâchées d'une erreur gaussienne sont effectuées tous les 10 points en espace et tous les 2 points en temps. L'ébauche est également une version bruitée par une loi normale du véritable signal initial $\mathbf{u}_0(s)$. Les matrices de covariances des erreurs d'ébauche et d'observations sont $\mathbf{B} = 0.1\mathbf{I}$ et $\mathbf{R} = 0.1\mathbf{I}$ respectivement.

Les deux types de signaux utilisés comme véritable état initial sont : un signal carré dont la dérivée est qualifiée de parcimonieuse et un signal carré avec des pentes moins abruptes dont la dérivée est qualifiée de quasi-parcimonieuse.

L'évolution du signal parcimonieux (ou creux) est représentée sur la figure (3.1) où nous pouvons voir ses fortes discontinuités. L'évolution du signal quasi-creux et ses discontinuités plus douces est représentée sur la figure (3.2). Les points rouges indiquent les observations faites lors des premiers pas de temps. Il y a donc 4 scénarios possibles au total : un scénario imparfait avec diffusion numérique et un scénario parfait sans diffusion basé sur le choix de μ, et pour chacun de ces deux scénarios nous considérons soit une condition initiale creuse soit quasi-creuse.

Deux types d'erreurs sont considérés pour analyser objectivement la qualité des résultats. La première est l'erreur quadratique moyenne (RMSE), définie par

$$RMSE(\mathbf{u}_t^a, \mathbf{u}_t^{tr}) = \frac{\| \mathbf{u}_t^a - \mathbf{u}_t^{tr} \|_2^2}{\| \mathbf{u}_t^{tr} \|_2^2}, \tag{3.10}$$

entre $\mathbf{u}_t^{tr}$ et $\mathbf{u}_t^a$, désignant respectivement le véritable état initial et l'état analysé intégrés jusqu'au temps t. La seconde est l'erreur relative absolue définie par

$$MAE(\mathbf{u}_t^a, \mathbf{u}_t^{tr}) = \frac{\| \mathbf{u}_t^a - \mathbf{u}_t^{tr} \|_1}{\| \mathbf{u}_t^{tr} \|_1}. \tag{3.11}$$

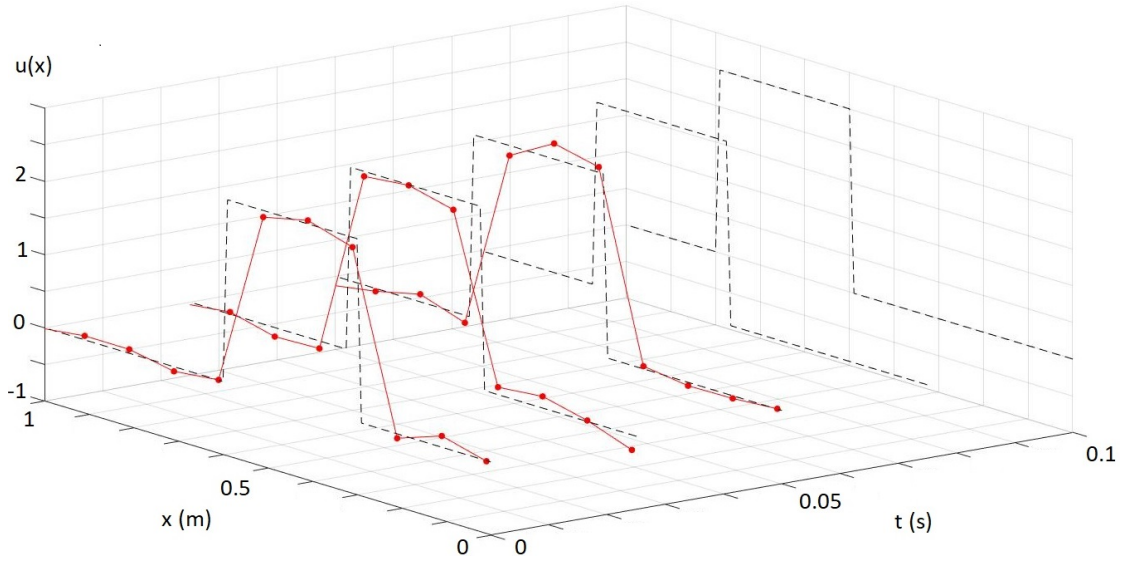


FIGURE 3.1 – Condition initiale parcimonieuses et son évolution temporelle. Un sous-échantillon des observations est représenté en rouge.

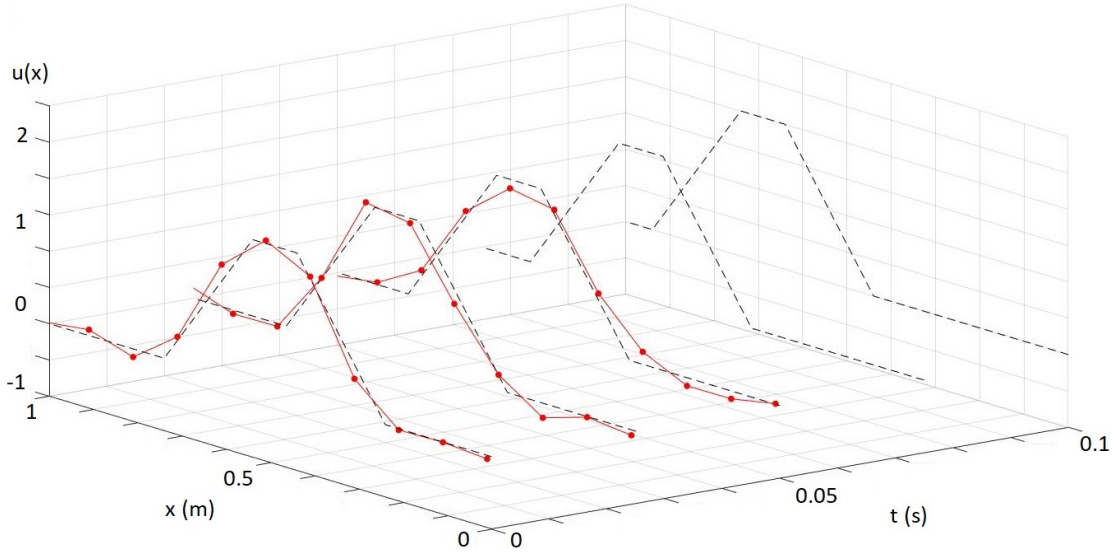


FIGURE 3.2 – Condition initiale quasi-parcimonieuse et son évolution temporelle. Un sous-échantillon des observations est représenté en rouge.

Pour pallier l'aléatoire intervenant dans les expériences (lors de la création de l'ébauche et des observations), chaque algorithme pour un scénario donné et avec une régularisation fixée sera répété 20 fois.

Enfin nous utiliserons comme algorithme de minimisation une simple descente de gradient. En effet, le 4DVar pénalisé par la norme L_p élevé à la puissance p est différentiable pour $p > 1$. De plus, l'objectif est ici simplement de montrer l'intérêt de la norme L_p en tant que régularisation, et la dimension faible du modèle-jouet nous permet de ne pas avoir à recourir à des algorithmes plus raffinés. La minimisation s'arrête lorsque le critère d'arrêt

$$\|\nabla\Omega_p(\mathbf{x}_k, \mathbf{b}, \lambda)\|_2 < 10^{-4}(\|\nabla\Omega_p(\mathbf{x}_0, \mathbf{b}, \lambda)\|_2 + \sqrt{\varepsilon_{machine}})$$

est vérifié. La fonctionnelle du 4DVar pénalisée avec la norme L_p s'écrit donc :

$$\Omega_p(\mathbf{x}) = \frac{1}{2}\|x - x_0^b\|^2_{B^{-1}} + \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{H}}x\|^2_{\mathbf{R}^{-1}} + \frac{\lambda}{p}\|\mathbf{\Phi x}\|_p^p$$

en gardant les notations de la section (1.2.2) et avec $\mathbf{\Phi}$ donnée par (3.4) ; la régularisation porte donc sur la dérivée numérique du signal. Nous pouvons écrire la fonctionnelle sous forme d'un moindre carrés pénalisé :

$$\Omega_p(\mathbf{x}) = \frac{1}{2}\|\mathbf{Ax} - \widetilde{\mathbf{b}}\|_2^2 + \frac{\lambda}{p}\|\mathbf{\Phi x}\|_p^p$$

Avec

$$\mathbf{A} = \begin{pmatrix} \mathbf{R}^{-\frac{1}{2}}\hat{\mathbf{H}} \\ \mathbf{B}^{-\frac{1}{2}} \end{pmatrix}; \quad \widetilde{\mathbf{b}} = \begin{pmatrix} \mathbf{R}^{-\frac{1}{2}}\mathbf{y} \\ \mathbf{B}^{-\frac{1}{2}}\mathbf{x}_b^0 \end{pmatrix},$$

ou encore, avec le changement de variable $\xi = \mathbf{\Phi x}$,

$$\Omega_p(\xi) = \frac{1}{2}\|\widetilde{\mathbf{A}}\xi - \widetilde{\mathbf{b}}\|_2^2 + \frac{\lambda}{p}\|\xi\|_p^p \tag{3.12}$$

avec

$$\widetilde{\mathbf{A}} = \begin{pmatrix} \mathbf{R}^{-\frac{1}{2}}\hat{\mathbf{H}} \\ \mathbf{B}^{-\frac{1}{2}} \end{pmatrix} \mathbf{\Phi}^{-1}.$$

Sans pénalisation, nous pouvons donc utiliser des solvers pour des problèmes aux moindres carrés standards (méthode QR etc.). Pour le cas particulier $p = 2$ on a directement

$$\Omega_2(\mathbf{x}) = \frac{1}{2}\| \begin{pmatrix} \mathbf{A} \\ \sqrt{\frac{\lambda}{2}}\mathbf{\Phi} \end{pmatrix} \mathbf{x} - \begin{pmatrix} \widetilde{\mathbf{b}} \\ \mathbf{0} \end{pmatrix} \|_2^2$$

que nous minimiserons de la même manière (avec la possibilité, cette fois, d'avoir une pénalisation).

Pour $p = 1$ nous retrouvons la pénalisation mixte en norme L_1/L_2 (en considérant que la fonctionnelle du 4DVar repose déjà sur une régularisation en norme L_2 des observations) proposée dans [33]. Nous écrivons alors le problème sous une forme quadratique avec contrainte, comme à l'équation (1.28), que l'on résout via la fonction quadprog de Matlab qui utilise un algorithme de points intérieurs.

3.2.2 Choix des paramètres λ et p

En accord avec ce qui a été présenté en section (1.3.4) nous commençons par tenter la méthode de la L-curve pour déterminer, à p fixé, un choix optimal de λ. L'exploration de différentes plages de valeurs pour λ à p fixé est cher en temps de calcul, et la courbe obtenue n'indique pas de valeurs claires pour le meilleur choix de λ. Elle permet néanmoins d'obtenir un ordre de grandeur des valeurs les plus pertinentes. À titre d'exemple, la figure (3.3) représente la courbe $(\log(\|\mathbf{A}\bar{\mathbf{x}}_{\lambda_j} - \mathbf{b}\|_p), \log(\|\mathbf{\Phi x}\|_p))$ pour $p = 1.2$ et λ appartenant à une liste de 100 valeurs réparties logarithmiquement également entre 10^{-2} et 10^3.

Pour un choix de λ moins cher en calculs et plus automatique, nous nous sommes tournés vers le principe de Morozov (présenté section 1.3.4). Afin que la fonctionnelle à minimiser soit de la même forme que celle présentée lors de l'introduction au principe de Morozov (1.33), nous utilisons l'équation (3.12) du 4DVar pénalisé.

Nous calculons λ par backtracking : en partant de $\lambda_0 = 100$ nous itérons $\lambda_{k+1} = 0.8\lambda_k$ jusqu'à obtenir

$$\|\widetilde{\mathbf{A}}\mathbf{x}_\lambda^\delta - \widetilde{\mathbf{b}}\| \leq \tau_2\delta. \tag{3.13}$$

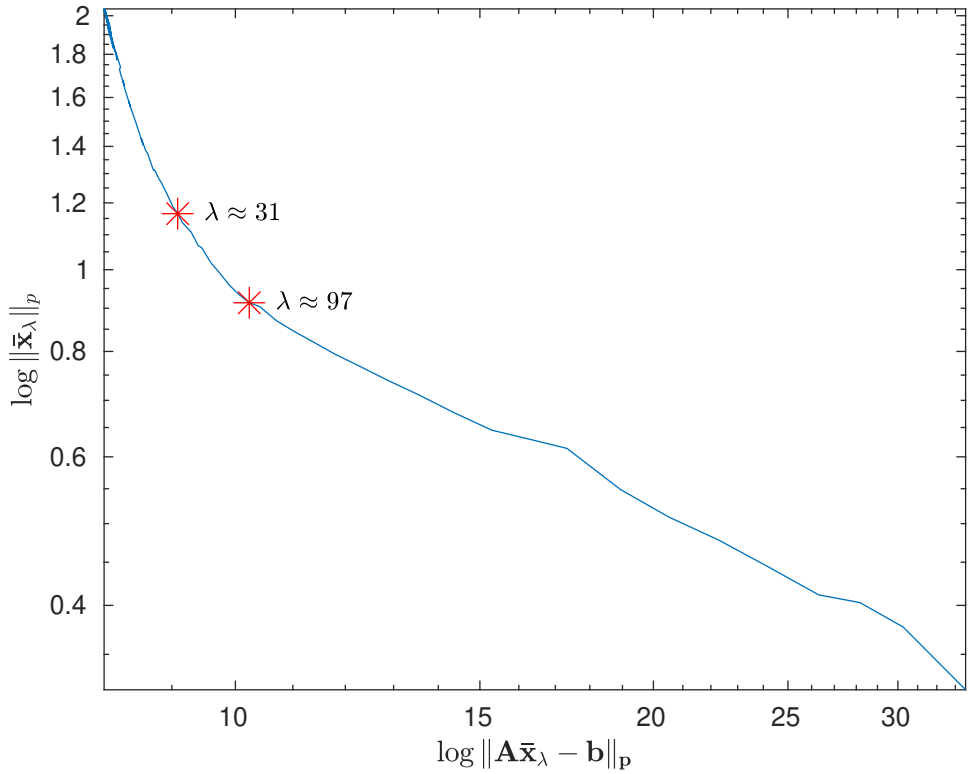


FIGURE 3.3 – L-curve pour $p = 1.2$ et λ allant de 10^{-2} à 10^3 en échelle logarithmique.

Le bruit ajouté au second membre est ici connu. Nous allons tout de même feindre l'ignorance et majorer l'erreur sur le second membre par $\delta = \sqrt{n}\sigma$ en prenant $\sigma = 1$ et $n = \text{taille}(\mathbf{y}) + \text{taille}(\mathbf{x}^b)$. Nous fixons de plus $\tau_2 = 1.1$. Par rapport à la définition du principe de Morozov, nous ne cherchons donc pas à vérifier la première inégalité $\|\widetilde{\mathbf{A}}\mathbf{x}_\lambda^\delta - \widetilde{\mathbf{b}}\| \geq \tau_1\delta$, ce qui engendrerait potentiellement plus de calculs. Le fait de procéder par backtracking et de retenir la plus grande valeur de λ vérifiant (3.13) permet néanmoins de ne pas sélectionner une valeur trop faible.

Dans notre cas, il existe aussi un degré de liberté supplémentaire sur le choix de p. En dehors de $p = 1$ et $p = 2$, nous considérerons les cas $p = 1.2$, $p = 1.5$ et $p = 1.8$ pour couvrir une plage de valeurs entre 1 et 2 puisqu'il s'agit d'une étude de l'impact numérique de la régularisation. Dans le cas introductif de l'épaisseur de la glace dans la mer de Beaufort, une valeur de p issue de la modélisation serait déjà disponible.

3.2.3 Résultats pour le scénario parfait

Nous commençons par considérer le scénario parfait ($\mu = 1$). Dans ce scénario, le modèle ne fait qu'une translation exacte de l'état entre l'instant t et l'instant $t+\Delta t$ selon l'équation d'advection (3.7). Par conséquent, la différence entre le signal reconstruit et le véritable état initial ne change pas au cours de l'évolution du modèle. En particulier, alors que le modèle imparfait va lisser les oscillations introduites dans la reconstruction de l'état initial, ce ne sera pas le cas pour ce scénario. Il suffit donc de comparer la RMSE et la MAE pour $t = 0$ entre l'état initial calculé par les différents algorithmes et le véritable état initial.

Un des défauts principaux de la fonctionnelle du 4DVar sans régularisation est l'apparition d'oscillations dans la reconstruction de l'état initial. Dans le but de réduire ces oscillations, la norme L_p, ici avec $p = 1, 1.2, 1.5, 1.8$ et $p = 2$, est rajoutée à la fonctionnelle. Les solutions obtenus en minimisant Ω_p pour différentes valeurs de p sont reportées sur la Figure (3.4). Nous notons que les oscillations décroissent en même temps que p décroît.

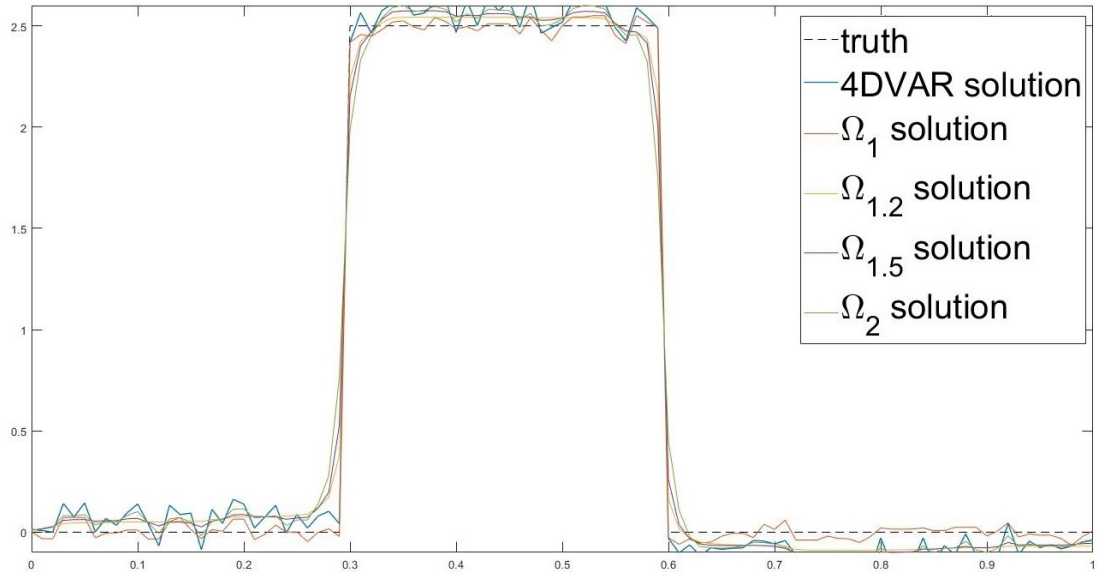


FIGURE 3.4 – Scénario parfait : véritable état initial (ligne pointillée en noire), et état analysé pour les différents formulations : 4DVar (ligne bleue), Ω_1 (ligne rouge), $\Omega_{1.2}$ (ligne jaune), $\Omega_{1.5}$ (ligne violette) et Ω_2 (ligne verte).

Tous les résultats concernant le scénario parfait sont collectés dans la Table (3.2). Pour aborder le problème de la robustesse de la régularisation, plusieurs choix de la variance du bruit ajouté à l'ébauche et aux observations ont été testés. Soulignons plusieurs conclusions émanant de cette table. D'abord, comme attendu pour un système d'assimilation

de données, les erreurs issues de la propagation de l'état analysé sont plus petites que celles issues de la propagation de l'ébauche, à l'exception de deux cas : la RMSE de l'algorithme minimisant Ω_2 pour le cas creux avec $\mathbf{B} = \mathbf{R} = 0.01\mathbf{I}$ et pour $\mathbf{B} = 0.01\mathbf{I}$ et $\mathbf{R} = 0.1\mathbf{I}$. De même, la précision de l'état analysé décroît lorsque les erreurs d'ébauche et d'observation croissent. La minimisation de la fonctionnelle du 4DVar classique Ω mène aux pires résultats globaux, suivi de la minimisation de Ω_2. Visiblement, la régularisation en norme L_2 n'est pas adaptée à la reconstruction d'un tel signal : trop d'oscillations sont créées.

Les quatre premières lignes du tableau (3.2) sont dominées par la fonctionnelle Ω_1, puis la RMSE et la MAE augmentent lorsque p augmente de 1 à 2. Les six dernières lignes sont, au contraire, dominées majoritairement par la fonctionnelle $\Omega_{1.2}$, que ce soit pour la RMSE ou la MAE. Pour $\mathbf{B} = 0.1\mathbf{I}$ et $\mathbf{R} = 0.01\mathbf{I}$, toutes les fonctionnelles Ω_p pour $p > 1$ battent Ω_1 par une une marge importante.

Nous analysons l'éventualité où aucune information sur l'ébauche n'est disponible sur les quatre dernières lignes de la Table (3.2). Le but d'une telle expérience étant de juger de la capacité de la régularisation à compenser un manque d'information vis-à-vis de l'erreur d'ébauche. Ainsi le 4DVar sans régularisation n'entre pas dans la comparaison et les fonctionnelles Ω_p ne contiennent plus de terme d'écart à l'ébauche. À nouveau la minimisation de Ω_1 mène aux meilleures RMSE et MAE. Paradoxalement, les erreurs sur l'état analysé obtenu par Ω_1 sont plus basses que celles obtenues par Ω_1 ou $\Omega_{1.2}$ avec un terme d'écart à l'ébauche (4 premières lignes du tableau). Il semble que sur cette expérience la régularisation en norme 1 supplée totalement à l'absence d'une ébauche bruitée. Au contraire, la minimisation de Ω_p pour $p > 1$ donne de moins bon résultats sans ce terme. Cela suggère que la régularisation en norme L_p avec $p > 1$ ne compense pas totalement un manque d'information a priori sur la structure de l'état initial.

Pour mieux visualiser l'aspect des différents états analysés, nous montrons sur la Figure (3.5) les résultats de vingt expériences lancées sur chaque fonctionnelle. Ici nous fixons $\mathbf{B} = 0.01\mathbf{I}$ et $\mathbf{R} = 0.01\mathbf{I}$. Le cas quasi-creux tend à rassembler les points et réduit fortement l'avance de la pénalisation en norme L_1 par rapport au cas creux. Bien que la minimisation de Ω_1 mène toujours aux meilleures erreurs moyennes, nous remarquons que le point qui obtient les meilleurs RMSE et MAE est issu de la minimisation de $\Omega_{1.2}$. Ces figures mettent aussi l'accent sur l'augmentation des erreurs qui accompagnent l'augmentation de p. En outre, elles montrent que les moyennes des erreurs présentées sur la Table (3.2) sont consistantes et ne dépendent pas fortement de l'aléatoire présent dans les expériences, ou d'une expérience particulière menant à des valeurs aberrantes.

Jusque là les fonctionnelles Ω_p avec $p = 1$, $p = 1.2$ et $p = 1.5$ semblent les plus prometteuses, selon les valeurs de $\mathbf{B}$ et $\mathbf{R}$ et pour un modèle parfait sans diffusion implicite. Maintenant, un vrai système d'assimilation de données est teinté d'erreurs (négligence de certains termes, erreurs numériques, erreurs modèles etc.) qui sont tout à fait à même

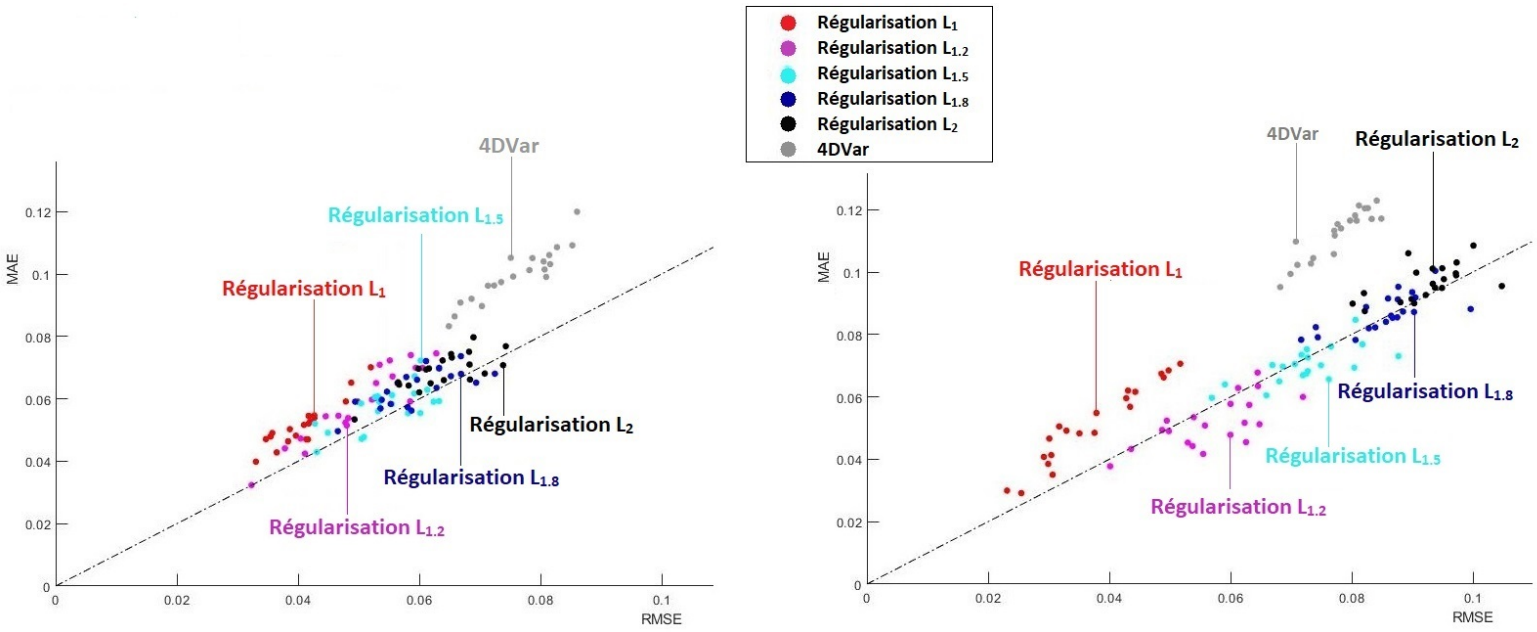


FIGURE 3.5 – RMSE et MAE de 20 expériences pour le scénario parfait avec $R = 0.1\mathbf{I}$ et $B = 0.1\mathbf{I}$. À gauche : le cas quasi-creux. À droite : le cas creux.

de faire perdre au signal sa sparsité initiale. Nous simulons ces erreurs par une diffusion implicite et investiguons les résultats obtenus pour ce nouveau scénario imparfait.

3.2.4 Résultats pour le modèle imparfait

Dans cette section nous prenons $\Delta t = 0.005$ pour avoir un nombre de Courant égal à 0.5. Nous fixons $\mathbf{B} = \mathbf{R} = 0.1\mathbf{I}$ excepté pour la Figure (3.10). Pour mieux visualiser l'impact de la diffusion implicite, nous montrons sur la Figure (3.6) le véritable état initial pour les deux conditions initiales creuse et quasi-creuse, et leur évolution lorsqu'elles sont soumises au modèle numérique imparfait. Ce sont les meilleures solutions que l'on peut obtenir dans le cadre d'un modèle imparfait lorsque l'erreur modèle n'est pas prise en compte dans la fonctionnelle du 4DVar. Nous remarquons que l'état initial est lissé avec le temps et perd sa parcimonie.

Regardons l'effet sur les états analysés dans ce nouveau cadre. Les résultats sont reportés sur les Figures (3.7) et (3.8). La solution du 4DVar sans pénalisation (« 4DVar solution ») concorde bien au véritable état initial et les discontinuités abruptes sont bien reconstruites. Cependant, comme précédemment, la solution présente de nombreuses oscillations. Ces oscillations sont atténuées par le modèle numérique au cours du temps (mais ne disparaissent pas complètement).

B	**R**	État initial		ébauche	$\mathbf{u}_{\text{4DVar}}$	$\mathbf{u}_{\text{4DVar,1}}$	$\mathbf{u}_{\text{4DVar,1.2}}$	$\mathbf{u}_{\text{4DVar,1.5}}$	$\mathbf{u}_{\text{4DVar,2}}$
0.01**I**	0.01**I**	Creux	RMSE	0.0918	0.0777	0.0373	0.0569	0.0724	0.0923
			RMAE	0.1338	0.1122	0.0513	0.0516	0.0700	0.0966
		Quasi-creux	RMSE	0.0885	0.0760	0.0413	0.0504	0.0545	0.0638
			RMAE	0.1184	0.0997	0.0519	0.0588	0.0573	0.0688
0.1**I**	0.1**I**	Creux	RMSE	0.2886	0.2427	0.1153	0.1667	0.1834	0.2063
			RMAE	0.4231	0.3530	0.1672	0.1980	0.2050	0.2343
		Quasi-creux	RMSE	0.2809	0.2388	0.1191	0.1370	0.1438	0.1568
			RMAE	0.3727	0.3119	0.1496	0.1771	0.1738	0.1845
0.01**I**	0.1**I**	Creux	RMSE	0.0901	0.0878	0.0748	0.0636	0.0804	0.1017
			RMAE	0.1331	0.1296	0.1089	0.0636	0.0828	0.1124
		Quasi-creux	RMSE	0.0875	0.0853	0.0742	0.0615	0.0668	0.0766
			RMAE	0.1174	0.1144	0.0983	0.0717	0.0694	0.0819
0.1**I**	**I**	Creux	RMSE	0.2799	0.2754	0.2376	0.2012	0.2177	0.2383
			RMAE	0.4129	0.4060	0.3479	0.2498	0.2456	0.2720
		Quasi-creux	RMSE	0.2769	0.2704	0.2319	0.1787	0.1818	0.1934
			RMAE	0.3664	0.3586	0.3042	0.2413	0.2293	0.2339
0.1**I**	0.01**I**	Creux	RMSE	0.2869	0.2014	0.2210	0.1141	0.1310	0.1475
			RMAE	0.4254	0.2619	0.3621	0.1028	0.1147	0.1393
		Quasi-creux	RMSE	0.2715	0.1904	0.2148	0.0852	0.0884	0.0973
			RMAE	0.3568	0.2233	0.3227	0.1044	0.0956	0.1059
-	0.01**I**	Creux	RMSE	0.0917	–	0.0359	0.1841	0.1866	0.1929
			RMAE	0.1339	–	0.0499	0.1971	0.1733	0.1809
		Quasi-creux	RMSE	0.0889	–	0.0406	0.1453	0.1439	0.1314
			RMAE	0.1174	–	0.0486	0.1908	0.1708	0.1402

TABLE 3.2 – Scénario parfait : RMSE et MAE pour les cas creux et quasi-creux. Le meilleur résultat de chaque ligne est souligné.

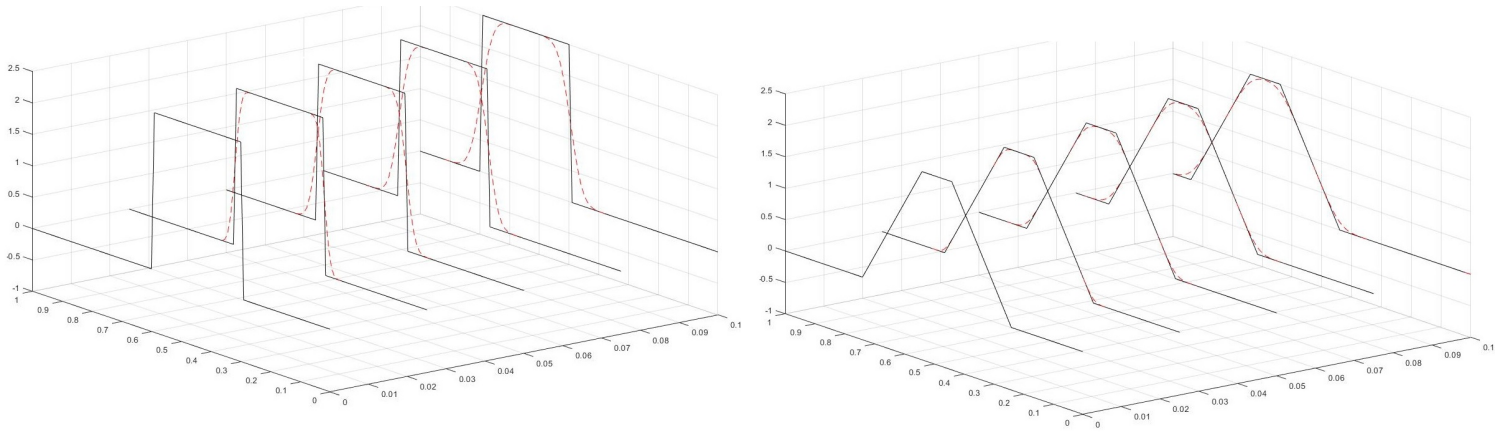


FIGURE 3.6 – L'évolution temporelle des vraies conditions initiales creuses (à gauche) et quasi-creuses (à droite) soumises au modèle imparfait est représentée par une ligne brisée rouge. L'évolution théorique de ces conditions soumises au modèle parfait est représentée par une ligne noire.

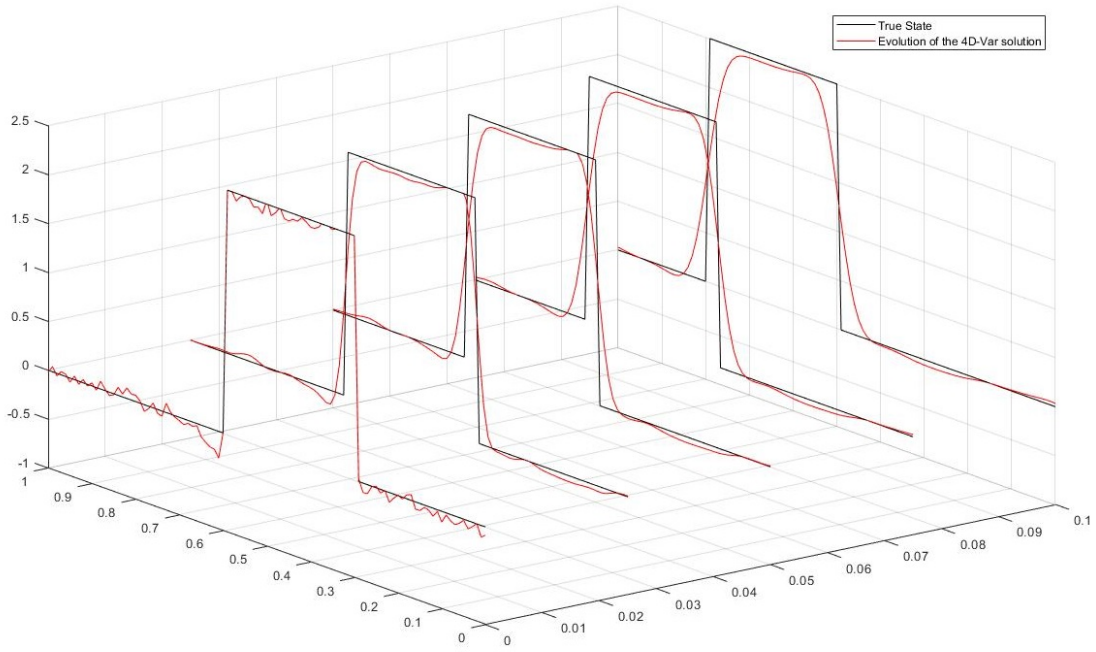


FIGURE 3.7 – Scénario imparfait : le résultat de la minimisation du 4DVar sans pénalisation et son évolution temporelle (cas creux). L'évolution du véritable état initial est représentée par une ligne noire. L'état analysé par le 4DVar classique est représenté par une ligne rouge.

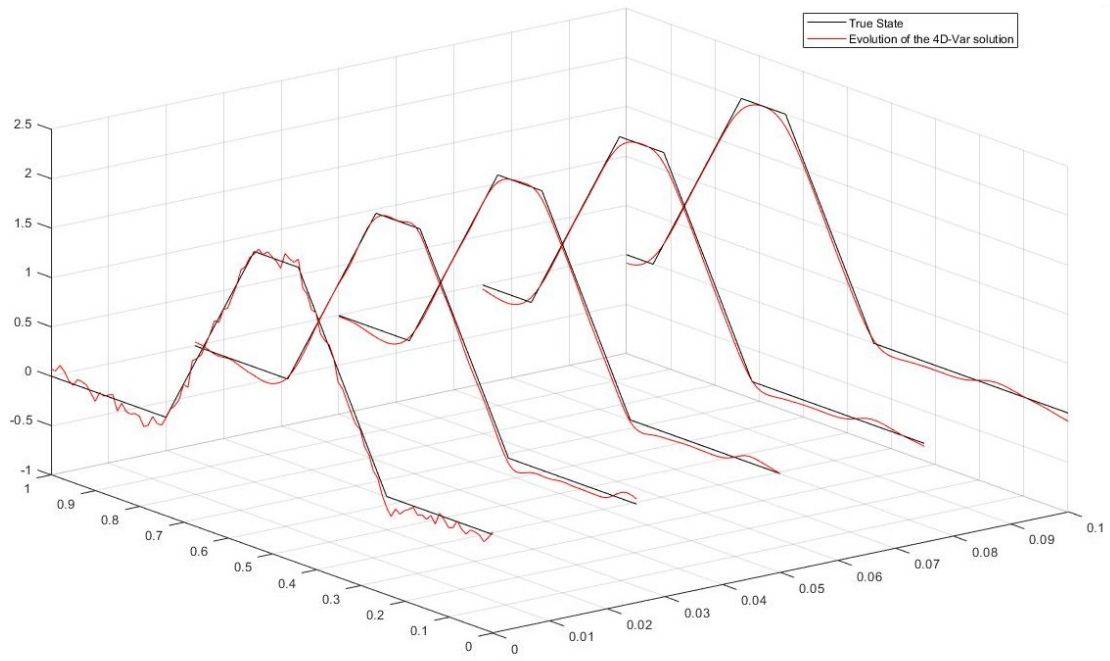


FIGURE 3.8 – Scénario imparfait : le résultat de la minimisation du 4DVar sans pénalisation et son évolution temporelle (cas quasi-creux). L'évolution du véritable état initial est représentée par une ligne noire. L'état analysé par le 4DVar classique est représenté par une ligne rouge.

Les erreurs des états analysés pour le cas creux sont comparées entre elles et à la propagation de l'ébauche à la Table (3.3). L'erreur modèle écarte les solutions du véritable état au cours du temps. En effet, les erreurs augmentent avec le temps lors de la propagation des vraies solutions initiales par le modèle.

$\mathbf{u}_t^a$	$t=0$		$t=0.025$		$t=0.05$		$t=0.075$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
imperfect	0	0	0.1526	0.0820	0.1841	0.1175	0.2047	0.1445
background	0.2943	0.4313	0.1947	0.2393	0.2112	0.2430	0.2261	0.2539
$\mathbf{u}_{4DVar}$	0.2746	0.3981	0.1738	0.1991	<u>0.1930</u>	0.2035	<u>0.2099</u>	0.2162
$\mathbf{u}_{4DVar,1}$	0.2431	0.3486	<u>0.1734</u>	0.1973	0.1931	<u>0.2032</u>	0.2101	<u>0.2161</u>
$\mathbf{u}_{4DVar,1.2}$	<u>0.1195</u>	<u>0.1331</u>	0.1876	<u>0.1808</u>	0.2119	0.2066	0.2290	0.2272
$\mathbf{u}_{4DVar,1.5}$	0.1454	0.1617	0.1946	0.1953	0.2154	0.2161	0.2305	0.2334
$\mathbf{u}_{4DVar,2}$	0.1754	0.2003	0.2037	0.2120	0.2202	0.2267	0.2334	0.2407

TABLE 3.3 – Scénario imparfait : RMSE et MAE pour le cas creux. Le meilleur résultat pour chaque colonne est souligné.

$\mathbf{u}_t^a$	$t=0$		$t=0.025$		$t=0.05$		$t=0.075$	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
vrai état initial	0	0	0.0410	0.0235	0.0660	0.0446	0.0863	0.0629
ébauche	0.2769	0.3662	0.1287	0.1718	0.1255	0.1642	0.1317	0.1673
$\mathbf{u}_{4DVar}$	0.2601	0.3428	0.1048	0.1398	<u>0.1010</u>	0.1310	<u>0.1085</u>	<u>0.1350</u>
$\mathbf{u}_{4DVar,1}$	0.1868	0.2408	<u>0.1019</u>	<u>0.1357</u>	<u>0.1010</u>	<u>0.1306</u>	0.1093	0.1355
$\mathbf{u}_{4DVar,1.2}$	<u>0.1313</u>	0.1686	0.1463	0.1837	0.1601	0.1979	0.1724	0.2109
$\mathbf{u}_{4DVar,1.5}$	0.1399	<u>0.1652</u>	0.1532	0.1782	0.1651	0.1904	0.1757	0.2017
$\mathbf{u}_{4DVar,2}$	0.1531	0.1774	0.1632	0.1861	0.1727	0.1950	0.1813	0.2039

TABLE 3.4 – Scénario imparfait : RMSE et MAE pour le cas quasi-creux. Le meilleur résultat pour chaque colonne est souligné.

Les évolutions temporelles des erreurs des états analysés obtenus avec les différentes régularisations sont illustrées sur la Figure (3.9). La principale différence entre les solutions survient au début de la fenêtre d'assimilation où plus la valeur de p augmente, plus les oscillations dites de Gibbs apparaissent. Cependant les solutions deviennent similaires lorsque t augmente à cause du lissage produit par le modèle imparfait, et se rapprochent de la solution de référence (« imperfect model » montré à la Figure (3.6)). Nous pouvons noter que dans la Table (3.3), la minimisation du 4DVar classique fournit la meilleure solution en terme de RMSE à partir du temps $t = 0.05s$, c'est-à-dire dans la seconde partie de la fenêtre d'assimilation. Cela est dû d'abord à l'utilisation de la norme L_2

pour le terme d'erreur. En effet, lorsque nous considérons la MAE, qui est plus sensible aux coefficients entre 0 et 1 (typiquement les faibles oscillations), les meilleurs résultats pour la seconde partie de la fenêtre d'assimilation sont obtenus par la minimisation de Ω_1. Cela est également dû à l'erreur modèle qui atténue l'importance de la parcimonie de l'état analysé lorsque que ce dernier est intégré dans le temps.

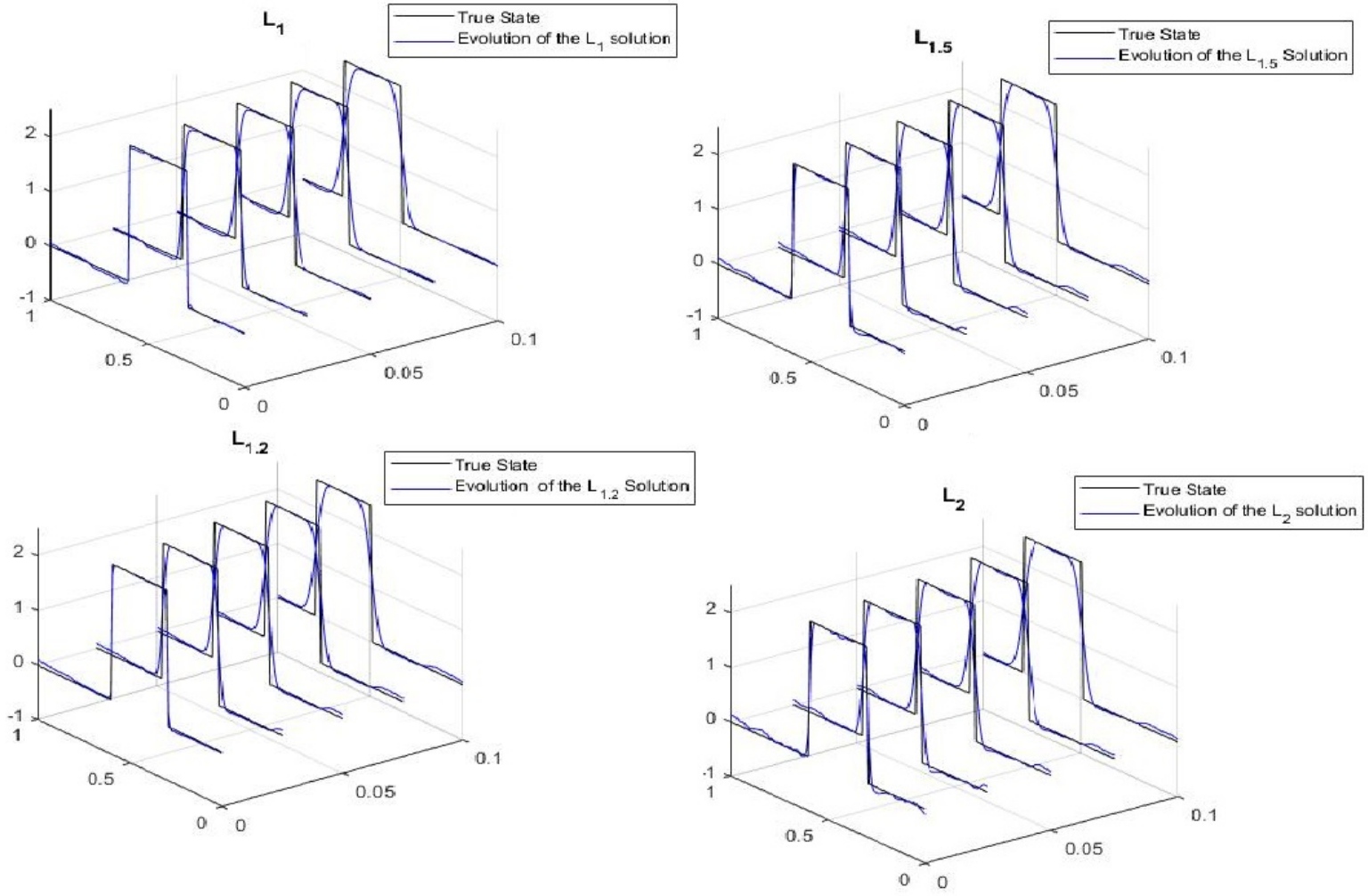


FIGURE 3.9 – Scénario imparfait : le résultat de la minimisation de Ω_1, $\Omega_{1.2}$, $\Omega_{1.5}$ et Ω_2 et son évolution dans le temps (en bleu). L'évolution du véritable état initial est représenté en noir.

De même, les résultats dans le cas d'un état initial quasi-creux sont reportés dans la Table (3.4). De nouveau, les meilleurs erreurs obtenues à $t = 0$ correspondent à la minimisation de $\Omega_{1.2}$ et $\Omega_{1.5}$. Pour mieux comprendre pourquoi la régularisation en norme

L_p, $1 < p < 2$ permet d'améliorer les résultats à l'instant initial, nous regardons plus attentivement l'effet de la régularisation sur la reconstruction sur la Figure (3.10). Dans un soucis de visualisation, nous prenons exceptionnellement $B = 0.9\mathbf{I}$ et $R = 0.01\mathbf{I}$ pour cette figure. La première partie de la figure montre les solutions obtenues avec les différentes régularisations. La ligne bleue correspond à $p = 1$, la ligne rouge à $p = 1.2$, la ligne jaune à $p = 1.5$ et la ligne violette à $p = 2$. Les parties (b) et (c) correspondent à un agrandissement des deux zones encadrées sur la partie (a). Nous remarquons que les oscillations sont fortement réduites par rapport à l'état analysé par le 4DVar classique. Cependant, les pentes ne sont pas bien reconstruites pour $p = 1$. En effet, l'état initial est ici, dans la base de $\mathbf{\Phi}$, non pas creux mais quasi-creux. De plus, la norme L_1 sélectionne des minimiseurs creux même lorsque la solution ne l'est pas. Ce comportement mène à des plateaux successifs au lieu d'une pente d'un seul tenant, visibles sur la partie (b) de la Figure (3.10). Cet « effet escalier »disparaît quand p devient strictement supérieur à 1. Mais des valeurs de p trop proches de 2 vont faire réapparaître des oscillations au niveau des zones plates (partie (c) de la Figure (3.10)). Ainsi les régularisations en norme $L_{1.2}$ et $L_{1.5}$ permettent d'obtenir les meilleurs RMSE et MAE à $t = 0$.

Mais lorsque le signal reconstruit est soumis au modèle, tous les états initiaux vont se rapprocher comme le montre la Figure (3.11). Les oscillations sont atténuées et la structure creuse de l'état initial est perdue (à cause de la diffusion numérique). À la fin de la fenêtre d'assimilation, c'est le 4DVar non pénalisé qui fournit les meilleurs RMSE et MAE. Les erreurs modèle vont compenser l'injection d'informations sur la structure de l'état initial (ce qui n'est pas le cas du scénario parfait). Ainsi la durée des fenêtres d'assimilation est critique vis-à-vis de la conservation de la structure du signal. Dans le cas de la diffusion numérique, des fenêtres d'une durée relativement courte doivent être utilisées pour ne pas perdre la parcimonie de l'état analysé.

3.2.5 Bilan de l'expérience d'advection

Nous avons vu l'impact de la régularisation en norme L_p, introduisant de l'information *a priori* sur la parcimonie de la solution sur un problème d'assimilation de données variationnelle.

L'efficacité de plusieurs régularisations ont été comparées dans un cadre d'expériences jumelles réalisées sur une équation d'advection 1D. Quatre cas ont été examinés : avec ou sans diffusion numérique et avec une condition initiale creuse ou quasi-creuse.

Nous avons constaté l'intérêt de la régularisation en norme L_p particulièrement dans le cas d'un signal quasi-creux, qui permet de faire un compromis entre la norme L_1 qui engendre « l'effet escalier » en surpénalisant les coefficients du signal, et la norme L_2 qui introduit des oscillations indésirables. La norme L_p avec $1 < p < 2$ a donc mieux permis de retrouver la structure du signal quasi-creux, bien que cette structure soit effacée avec le temps dans le cas du scénario imparfait avec diffusion numérique, suggérant d'utiliser

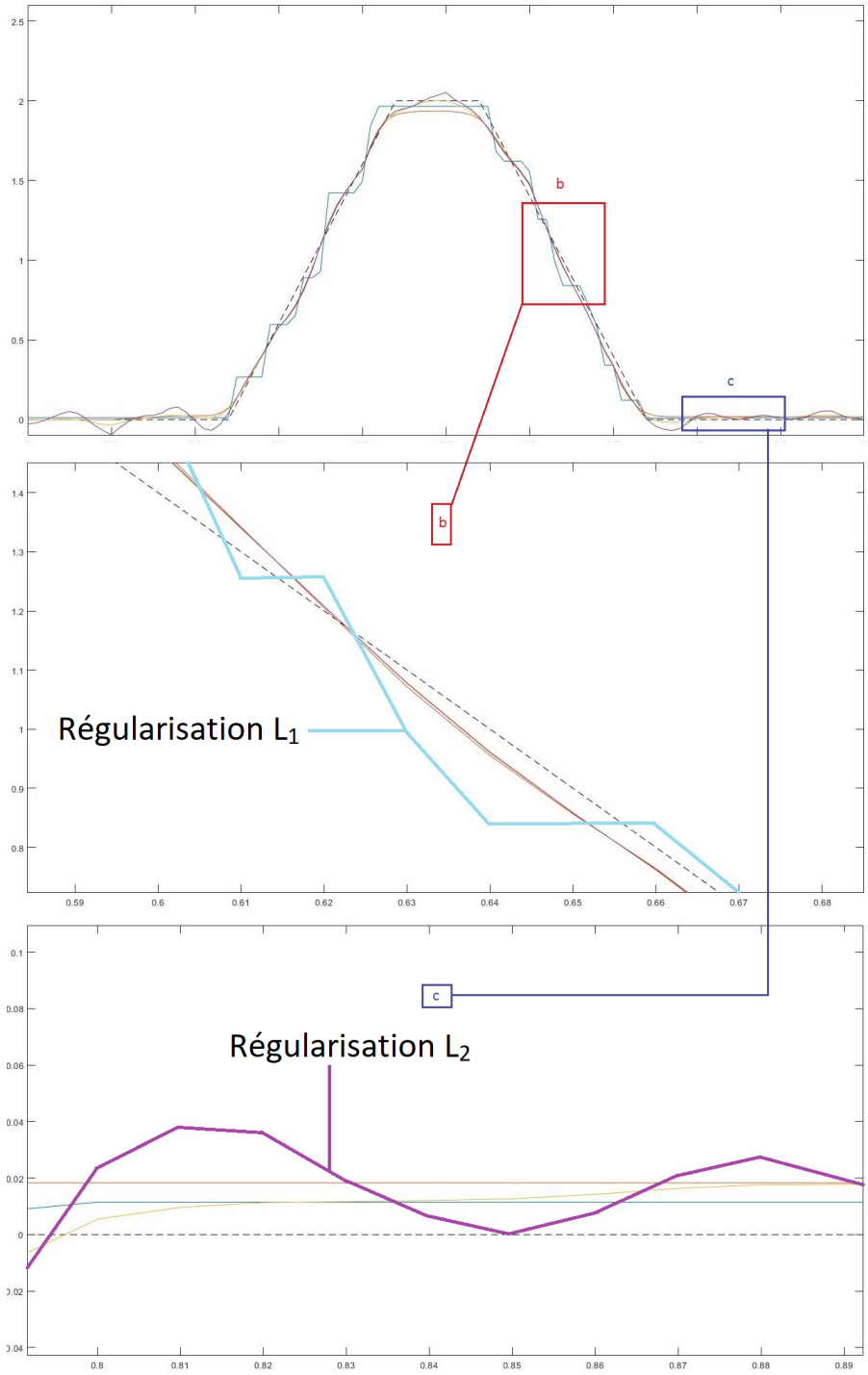


FIGURE 3.10 – Scénario imparfait : vrai état initial quasi-creux (ligne noire pointillée) et état analysé obtenu en minimisant Ω_1 (ligne bleue), $\Omega_{1.2}$ (ligne rouge), $\Omega_{1.5}$ (ligne jaune) and Ω_2 (ligne violette). Ici $B = 0.9\mathbf{I}$ et $R = 0.01\mathbf{I}$.

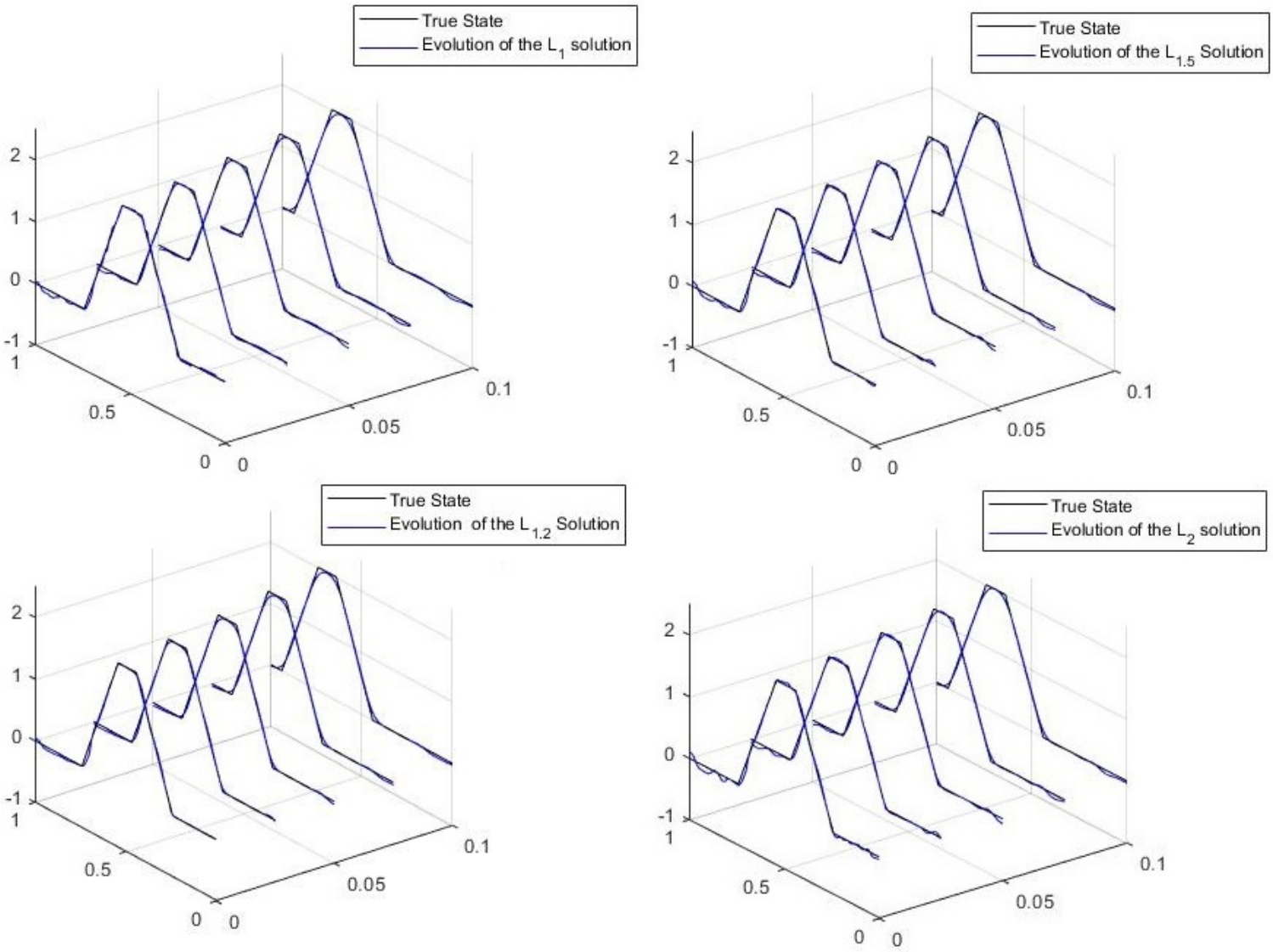


FIGURE 3.11 – Scénario imparfait : état analysé pour Ω_1, $\Omega_{1.2}$, $\Omega_{1.5}$ et Ω_2 représentés par une ligne bleue. L'évolution du vrai état initial par l'advection linéaire est représenté par une ligne noire.

des fenêtres d'assimilation de courte durée. Globalement, les régularisation en norme $L_{1.2}$ et $L_{1.5}$ sont plus efficaces quand l'incertitude sur l'ébauche et les observations augmentent et pallient ainsi un manque d'informations d'une autre nature. Enfin, si la régularisation en norme L_1 reste la plus efficace dans le cas d'un signal creux sans diffusion numérique, la norme L_p avec p légèrement supérieur à 1 est capable de rivaliser avec elle en terme d'erreurs de l'état analysé, tout en gardant un problème d'optimisation différentiable sans contraintes.

3.3 Conclusion

Nous avons vu que la régularisation en norme L_p découlait naturellement des hypothèses statistiques faites sur la variable d'état. Mais, même sans ces hypothèses, les expériences de ce chapitre ont permis de mettre en exergue un cadre d'application favorable à cette régularisation : le cas des signaux quasi-creux. Les résultats sur le problème d'advection 1D, proxy d'un système d'assimilation plus avancé, sont encourageants pour appliquer cette régularisation dans le cas de la concentration de la glace dans la mer, le cas de la concentration de la sargasse / du phytoplancton dans les océans ou l'étude de fronts météorologiques présentant ce type de structure quasi-creuse.

Le réglage du coefficient de régularisation λ et de la valeur de p reste problématique dans le cas général. Se pose avant tout la question de minimiser efficacement la fonctionnelle du 4DVar pénalisée avec une norme L_p. En effet une simple descente de gradient converge lentement et son implémentation dans un cadre réaliste ne serait pas idoine. De plus, les algorithmes plus évolués (tel que RPCG [59]) n'ont pas été conçus pour prendre en compte ce type de pénalisation.

Chapitre 4

Algorithmes de minimisation du 4DVar pénalisé en norme L_p

4.1 Motivations pour effectuer la minimisation dans un espace non euclidien . 86
4.2 Comparaison théorique des différentes descentes de gradient dans les espaces de Banach . . . 87
4.2.1 Panorama des algorithmes . . . 87
4.2.2 Descente de gradient dans le dual avec recherche linéaire . . . 88
4.2.3 Gradient conjugué non linéaire dans l'espace dual . . . 93
4.2.4 Gradient conjugué non linéaire avec transport de la direction dans le primal . . . 97
4.3 Comparaisons expérimentales des algorithmes . . . 100
4.3.1 Comparaison par rapport au choix de β_k . . . 100
4.3.2 Comparaison des vitesses de convergence . . . 103
4.4 Convergence des algorithmes . . . 106
4.4.1 Algorithme de descente de gradient dans l'espace dual . . . 106
4.4.2 Algorithme de gradient conjugué non linéaire dans le dual . . . 110
4.4.3 Algorithme du gradient conjugué non linéaire avec transport de la direction dans le primal . . . 111
4.5 Conclusion . . . 114

Ce chapitre présente les algorithmes que nous proposons pour minimiser le 4DVar avec un terme de pénalisation en norme L_p. Leur principale particularité est qu'ils sont conçus pour la minimisation de fonctionnelles dans des espaces de Banach. Nous commençons par justifier le choix de ce cadre mathématique puis nous exposons un algorithme de descente de gradient dans l'espace dual dont la recherche de pas s'effectue également dans l'espace dual, avant d'adapter l'algorithme du gradient conjugué non linéaire afin d'opérer également ses itérations dans l'espace dual.

Par soucis de clarté nous abandonnons ici les notations de l'assimilation de données : les variables (resp. les opérateurs) ne seront plus en gras pour indiquer si ce sont des vecteurs (resp. des opérateurs linéaires).

4.1 Motivations pour effectuer la minimisation dans un espace non euclidien

Un choix suggéré par la physique et la régularisation...

Premièrement, dans le cas d'une modélisation d'un problème physique continu, ajouter une pénalisation en norme L_p n'a de sens que si la variable pénalisée appartient à l'espace L_p lui-même. Or nous avons vu que cette pénalisation pouvait être justifiée de plusieurs manières :

— par la modélisation statistique du problème

— par l'intérêt numérique de la pénalisation.

Pour nos applications, la fonctionnelle du 4DVar est déjà une formulation discrétisée sur une grille finie du problème considéré. La variable d'état $\mathbf{x}$ appartient donc à $\mathbb{R}^n$. Nous souhaiterions équiper cet espace de la même norme que précédemment dans le cas continu. Équipé de la norme euclidienne classique, $(\mathbb{R}^n, \|\cdot\|_2)$ est un espace de Hilbert. Autrement, il s'agit d'un espace non euclidien. Nous utilisons également l'appellation « d'espace de Banach », que l'espace soit de dimension finie ou non, par analogie avec les espaces L_p ou l_p puisque nous munissons $\mathbb{R}^n$ d'une norme L_p (non issue d'un produit scalaire pour $p \neq 2$).

Parallèlement, dans certains cas l'espace L_p est directement le cadre le plus naturel pour traiter certains problèmes physiques tel que celui de la dynamique de fluides non newtoniens ou celui du traitement d'images qui font intervenir l'opérateur p-Laplacien

$$\Delta_p \mathbf{x} := \nabla \cdot (|\nabla x|^{p-2} \nabla \mathbf{x})$$

en ayant noté

$$|\nabla \mathbf{x}|^{p-2} = \left[\left(\frac{\partial \mathbf{x}}{\partial x_1} \right)^2 + \cdots + \left(\frac{\partial \mathbf{x}}{\partial x_n} \right)^2 . \right]^{\frac{p-2}{2}}$$

Dans tous les cas il est donc raisonnable de penser que l'opérateur A du 4DVar pénalisé (4.1) réécrit ci-dessous est un opérateur de L_p (ou de l_p) dans $\mathbb{R}^m$.

$$\|A(\mathbf{x}) - \mathbf{b}\|_2^2 + \lambda \|\mathbf{\Phi x}\|_p^p. \tag{4.1}$$

Ainsi il est tentant d'essayer de minimiser (4.1) par des algorithmes de minimisation dans les espaces de Banach (bien qu'une méthode classique de minimisation d'une fonctionnelle différentiable marcherait également).

... et conforté par l'expérimentation

Les raisons précédentes poussent à essayer une descente de gradient dans un espace de Banach sur l'expérience d'advection linéaire du chapitre précédent, où une descente de gradient simple suffisait pour minimiser (4.1). Nous allons illustrer que, en effet, les résultats obtenus encouragent à attaquer des problèmes de grande dimension avec un terme de pénalisation en norme L_p via ce type d'algorithmes.

Comme vu au premier chapitre, il existe plusieurs manières de faire une descente de gradient dans un espace de Banach. Nous nous attelons maintenant à comparer les divers schémas de descente de gradient dans les espaces de Banach pour sélectionner le plus pertinent, montrer théoriquement sa convergence et s'en inspirer pour adapter l'algorithme du gradient conjugué non linéaire.

4.2 Comparaison théorique des différentes descentes de gradient dans les espaces de Banach

4.2.1 Panorama des algorithmes

Résumons les différentes méthodes de gradient possibles pour minimiser le $4DVar$ pénalisé par une norme L_p (4.1), et dont nous disposons de la convergence théorique :

1. Algorithmes avec une boucle interne de linéarisation (de type Gauss-Newton).
2. Descente de gradient classique.
3. Descente de gradient avec transport des itérés dans le dual, avec pas donné par l'algorithme de Boneski *et al.* (2.25)
4. Descente de gradient avec transport de la direction dans le primal et pas obtenu par une recherche linéaire.

La méthode 3. converge pour une recherche linéaire le long de la direction transportée dans le primal donnant un pas α_k vérifiant

$$f_{k+1} \leq f_k + c\alpha_k \|\nabla f_k\|_q^q \text{ avec } c \in]0;1[.$$

Boneski *et al.* ont remarqué ([71]) que le schéma 4. convergeait plus rapidement (en nombre d'itérations) que le schéma 3. En effet le pas du schéma 3 est construit de manière *ad hoc* pour obtenir la convergence théorique d'une part, et d'autre part repose sur l'estimation de $\sup\{\|\psi_k\|^2 : \psi \in \partial\Psi(x)\}$ qui n'est pas trivial en général et qui peut drastiquement réduire la taille du pas si celle-ci est surestimée. Nous souhaitons donc également utiliser une recherche linéaire du pas lorsque les itérés sont transportés dans le dual ; c'est le premier algorithme que nous proposons.

Complexité en espace et en calcul

Tous les algorithmes présentés ici ne nécessitent que l'évaluation de la fonction et de son gradient, notamment lors de la recherche de pas. Les algorithmes faisant usage de l'espace dual se distinguent par l'utilisation de l'opérateur de dualité J dont l'évaluation en un vecteur $\mathbf{x} \in \mathbb{R}^n$ requiert $\mathcal{O}(n)$ opérations, qui est négligeable devant les opérations précédemment citées. En outre, ces algorithmes restent totalement compatibles avec des routines capables d'évaluer un opérateur linéaire en un point sans stocker explicitement l'opérateur sous forme matricielle. Ils héritent donc des complexités en espace et en calcul de leurs algorithmes parents qui sont celui de descente de gradient et celui de gradient conjugué non linéaire.

4.2.2 Descente de gradient dans le dual avec recherche linéaire

Nous gardons le schéma d'une descente de gradient avec transport des itérés dans l'espace dual :

$$p_0 = -f'_0, \tag{4.2}$$
$$x^*_{k+1} = x^*_k + \alpha_k p_k, \tag{4.3}$$
$$x_{k+1} = J_q(x^*_{k+1}), \tag{4.4}$$
$$p_{k+1} = -f'_{k+1}. \tag{4.5}$$

Pour éviter d'avoir un nouveau paramètre variable, et pour simplifier les calculs, nous prenons dorénavant pour fonction de jauge (de la définition de l'opérateur de dualité J_r^X (2.2.7) la fonction $t \to t^r$ avec r **l'indice de la norme qui équipe l'espace** X. Ainsi pour $X = L_p$ ou $X = (\mathbb{R}^n, \|\cdot\|_p)$ la fonction de jauge sera $t \to t^p$ et l'opérateur de dualité de X sera noté J_p. En particulier nous avons

$$J_p(x) = (|x_1|^{p-1}\,\mathrm{signe}(x_1), \cdots, |x_n|^{p-1}\,\mathrm{signe}(x_n)).$$

Pour $p > 2$, $x \to |x|^{p-1}\mathrm{signe}(x)$ est différentiable sur $\mathbb{R}$ et nous avons également :

$$J'_p(x) = (p-1)\begin{pmatrix} |x_1|^{p-2} & 0 & \cdots & 0 \\ 0 & |x_2|^{p-2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & |x_n|^{p-2} \end{pmatrix} \tag{4.6}$$

En pratique nous utiliserons J'_q avec q l'exposant conjugué de p. Comme nous considérons $1 < p < 2$, nous avons $q \geq 2$, et J'_q est bien défini.

Transporter les itérations de $(\mathbb{R}^n, \|\cdot\|_p)$ vers $(\mathbb{R}^n, \|\cdot\|_q)$ par l'opérateur J_p peut être vu comme effectuer un préconditionnement non linéaire à droite par l'opérateur J_q, ce qui peut pousser à remplacer l'étape (4.5) par $p_{k+1} = -(f \circ J_q)'(x^*_{k+1})$. Il y a alors un risque de se diriger vers un point critique de $(f \circ J_q)$ et non de f. Les points critiques de $f \circ J_q$ sont caractérisés par un voisinage très plat comme l'illustre la Figure (4.1). Sur cette figure, nous regardons une simple quadratique en deux dimensions donnée par $z = (x-1)^2 + (y-1)^2$ avant et après le changement de variable $(x', y') = J_q(x, y) = (\text{signe}(x)|x|^{q-1}, \text{signe}(x)|x|^{q-1})$. Même si une stratégie pour éviter cette situation est explorée par la suite, on peut gâcher des itérations en se dirigeant vers ces points, voire ne pas pouvoir en sortir. C'est le cas en considérant par exemple la minimisation des moindres carrés $\|\mathbf{Ax} - \mathbf{b}\|_2^2$, avec $\mathbf{A} = \begin{pmatrix} 0.7156 & 0.7417 & 0.5250 \\ 0.8007 & 0.0191 & 0.4633 \\ 0.7065 & 0.8860 & 0.0652 \end{pmatrix}$; $b = \begin{pmatrix} 0.7134 \\ 0.4889 \\ 0.6677 \end{pmatrix}$; $x_0 = \begin{pmatrix} 1000 \\ 1000 \\ 1000 \end{pmatrix}$ (matrices et vecteurs générés aléatoirement) ; $p = 1.2$ (donc $q = 6$) ; $\lambda = 0$. L'algorithme avec $p_k = -\nabla f_k$ va bien vers la solution du système $\bar{\mathbf{x}} = \begin{pmatrix} 0.4866 \\ 0.3509 \\ 0.2000 \end{pmatrix}$ tandis que celui avec $p_k = -(f \circ J_q)'(x^*_{k+1})$ est arrivé au point $\mathbf{x} = \begin{pmatrix} -0.0005 \\ 0.5857 \\ 0.7643 \end{pmatrix}$ et s'est arrêté à cause d'une stagnation autour de ce point : $\|x_{k+1} - x_k\|_2 \leq 10^{-12}(\|x_k\|_2 + \sqrt{\varepsilon})$

De plus, cette direction est plus sensible numériquement car si p est proche de 1, q est grand et les composantes de $J'_q(x_{k+1})$ en seront d'autant amplifiées (plus de détails sur la limitation numérique du choix de p sont donnés à la section 2.1.2).

Notons que pour tout $p_k \in X^*$,

$$\begin{aligned} \langle \nabla(f \circ J_q)(x^*_k), p_k \rangle &= \langle J'_q(x^*_k)^T \nabla f(J_q(J_p(x_k)), p_k \rangle \\ &= \langle J'_q(x^*_k) \nabla f(x_k), p_k \rangle \\ &= \langle \nabla f(x_k), J'_q(x^*_k) p_k \rangle. \end{aligned}$$

Ainsi si p_k est une direction de descente pour $f \circ J_q$ alors $J'_q(x^*_k)p_k$ est une direction de descente pour f. Les fonctions f et $f \circ J_q$ n'ont pas les mêmes points critiques : on peut néanmoins avoir $\nabla f_k^T J'_q(x^*_k)p_k = 0$ pour un point x_k sans que $\nabla f_k = 0$ (lorsque x_k possède des composantes, nulles d'après l'expression (4.6)). Après avoir présenté les conditions de recherche linéaire dans l'espace dual, nous y ajouterons un garde-fou pour éviter de converger théoriquement vers un point critique de $f \circ J_q$ qui n'est pas un point critique de f.

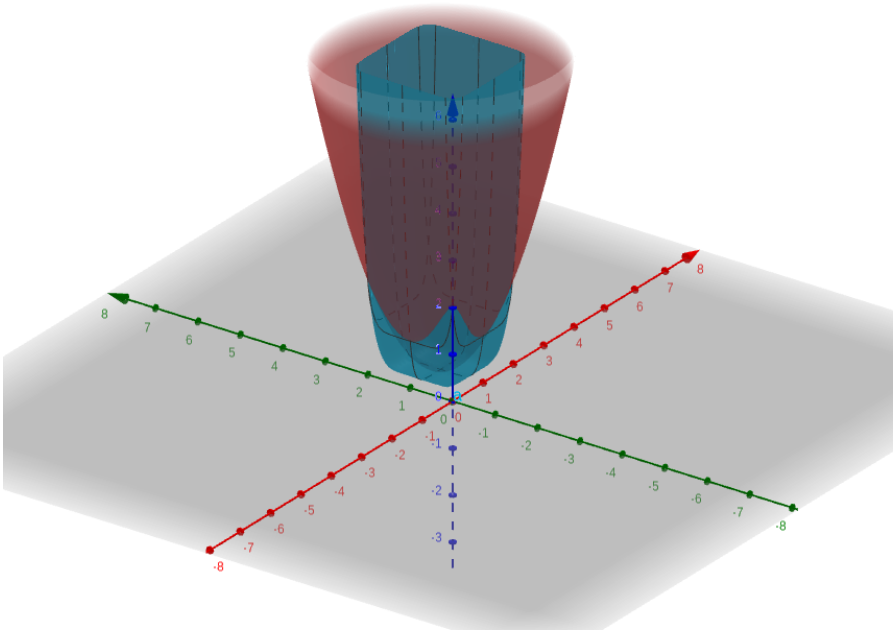


FIGURE 4.1 – Quadratique avant (en rouge) et après (en bleu) le changement de variable $(x', y') = J_q(x, y) = (\text{sgn}(x)|x|^{q-1}, \text{sgn}(y)|y|^{q-1})$. Les pentes sont plus fortes après changement de variable, et la zone autour du point critique plus plate, à cause des exposants (ici $p = 1.2$ / $q = 6$).

Condition d'Armijo dans l'espace dual

Nous adaptons à présent les conditions des recherches linéaires pour qu'elles s'effectuent dans l'espace dual. La condition d'Armijo (2.1.1) écrite pour $f \circ J_q$ devient :

$$(f \circ J_q)(x_k^* - \gamma^\nu p_k) \leq f_k - c_1 \gamma^\nu \nabla f_k^T J_q'(x_k^*) p_k, \tag{4.7}$$

avec $0 < \gamma < 1$ et ν le plus grand entier naturel calculé par backtracking rendant l'inégalité (4.7) vraie.

Conditions de Wolfe dans l'espace dual

Les conditions de Wolfe adaptées à $f \circ J_q$ deviennent :

$$\begin{aligned} f_{k+1} = (f \circ J_q)(x_k^* + \alpha_k p_k) &\leq f_k + c_1 \alpha_k \nabla f_k^T J_q'(x_k^*) p_k, \\ \nabla f_{k+1}^T J_q'(x_k^*) p_k &\geq c_2 \nabla f_k^T J_q'(x_k^*) p_k. \end{aligned}$$

Rappelons que vouloir vérifier la seconde condition de Wolfe peut coûter très cher en pratique à cause de l'évaluation du gradient de f en plusieurs points. Cette condition permet néanmoins de faire des pas plus grands ; son utilisation dépend donc de la difficulté à évaluer ∇f_k.

Pour converger vers un point qui est bien un point critique de f (et non pas de $f \circ J_q$ seulement) nous modifions ces conditions en y ajoutant un « garde-fou ».

garde-fou pour les recherches linéaires

L'idée est de remplacer l'argument de $J_q'(x_k^*)$ par un autre vecteur v_k tel que si $\nabla f_k^T J_q'(v_k)p_k = 0$, alors on aurait quand même $\nabla f_k = 0$. Puisqu'en pratique nous prenons comme direction $p_k = -\nabla f_k$, nous nous sommes tournés vers le choix $v_k = \nabla f_k$.

Numériquement nous fixons donc $\varepsilon > 0$, et nous introduisons un opérateur H_k envoyant x_k^* sur ∇f_k lorsque $|\nabla f_k^T J_q'(x_k^*)p_k|$ est proche de zéro :

$$H_k(x_k^*) = \begin{cases} \nabla f_k & \text{si } |\nabla f_k^T J_q'(x_k^*)p_k| < \varepsilon \\ x_k^* & \text{sinon} \end{cases} \tag{4.8}$$

La condition d'Armijo dans le dual avec ce garde-fou s'écrit alors :

$$(f \circ J_q)(x_k^* - \gamma^\nu p_k) \leq f_k - c_1\gamma^\nu \nabla f_k^T J_q'(H_k(x_k^*))p_k, \tag{4.9}$$

et celles de Wolfe deviennent :

$$f_{k+1} = (f \circ J_q)(x_k^* + \alpha_k p_k) \leq f_k + c_1\alpha_k \nabla f_k^T J_q'(H_k(x_k^*))p_k \tag{4.10}$$

$$\nabla f_{k+1}^T J_q'(H_k(x_k^*))p_k \geq c_2 \nabla f_k^T J_q'(H_k(x_k^*))p_k. \tag{4.11}$$

Lorsque $H_k(x_k^*) = x_k^*$ un tel pas α_k existe car (4.10) et (4.11) sont les conditions de Wolfe standards pour la fonctionnelle $f \circ J_q$. Ce pas peut toujours être calculé par un algorithme de bissection.

Lorsque $H_k(x_k^*) = \nabla f_k$, les conditions (4.10) et (4.11) peuvent être vues comme des conditions de Wolfe pour f que l'on ferait décroître dans la direction $J_q'(\nabla f_k)p_k$. L'existence de α_k est assurée dès que $J_q'(\nabla f_k)p_k$ est une direction de descente pour f. C'est bien le cas pour $p_k = -\nabla f_k$ puisque, $J_q'(x)$ étant semi-définie positive pour tout x, on a

$$\nabla f_k^T J_q'(\nabla f_k)(-\nabla f_k) \leq 0$$

avec égalité si et seulement si $\nabla f_k = 0$.

Remarque 3. *Dans le cas général où p_k est quelconque, plusieurs choix existent pour $H_k(x_k^*)$ lorsque $|\nabla f_k^T J_q'(x_k^*)p_k| < \varepsilon$, de manière à ce que $J_q'(H_k(x_k^*))p_k$ reste une direction de descente pour f dès que p_k en est une. Le choix le plus simple est de prendre $H_k(x_k^*) = \mathbf{1}$ le vecteur dont toutes les composantes valent 1. À ce moment $J_q'(H_k(x_k^*)) = \mathbf{I}$ et on retrouve les conditions de Wolfe standards.*

Remarque 4. *Une autre piste avait été explorée initialement pour converger vers un point critique de f malgré les conditions de Wolfe portant sur $f \circ J_q$. Pour éviter le problème des composantes nulles de x_k^* (ou de x_k) qui engendreraient $J_q'(x_k^*)\nabla f_k = 0$ alors que $\nabla f_k \neq 0$, nous voulons introduire ϵ_k une matrice diagonale dont le i^{eme} élément diagonal est défini par*

$$\epsilon_k^i = \begin{cases} \epsilon & si\ |x_k|^i < \epsilon \\ 0 & sinon \end{cases}$$

et nous modifions les conditions de Wolfe par :

$$f_{k+1} = (f \circ J_q)(x_k^* + \alpha_k p_k) \leq f_k + c_1 \alpha_k \nabla f_k^T (J_q'(x_k^*) + \epsilon_k)p_k \tag{4.12}$$

$$\nabla f_{k+1}^T (J_q'(x_k^*) + \epsilon_k)p_k \geq c_2 \nabla f_k^T (J_q'(x_k^*) + \epsilon_k)p_k. \tag{4.13}$$

Si ces dernières conditions sont vérifiées, nous pouvons montrer qu'une descente de gradient dans le dual converge également. Cependant l'existence pour tout k d'un pas α_k telle que (4.12) soit vérifiée n'est pas garantie (ni, a fortiori, l'existence d'un pas vérifiant simultanément (4.12) et (4.13)). La même remarque s'applique donc à la condition d'Armijo modifiée de la même manière.

Sensibilité numérique par rapport à p

On peut chercher à prendre p le plus proche de 1 pour réduire les oscillations de Gibbs ou simplement s'approcher de la norme L_1 - voir section (1.3.3). Les valeurs de $x^* = J_p(x) = \text{Signe}(x)|x|^{p-1}$ peuvent alors se heurter à la précision machine et renvoyer $x^* = x$ sans que x ne soit solution de l'équation $J_p(x) = x$. Soit N le plus grand nombre en virgule flottante représentable sur l'ordinateur : $N_{double} \approx$ 1.8e+308 en double précision et $N_{simple} \approx$ 3.4e+38 en simple précision. Si $p-1$ est trop proche trop de 0 nous risquons donc d'avoir $x^{p-1} = 1$, ou, dit autrement, si $\frac{1}{p-1}$ est trop grand, nous risquons d'avoir $x^{\frac{1}{p-1}} = +\infty$. Cela se produit dès que $x^{\frac{1}{p-1}} \geq N$.

Soit également M une borne supérieure de la magnitude des composantes de x, pour x appartenant à un ensemble borné. Ce choix doit être guidé par l'intuition physique : par exemple, si $\mathbf{x}$ décrit la hauteur du niveau de la mer, nous pouvons facilement donner un majorant. Nous supposons que $M \geq 1$, et nous souhaitons que

$$M^{\frac{1}{p-1}} \leq N.$$

p doit donc être tel que

$$p \geq \frac{\log M}{\log N} + 1.$$

Pour $M = 10^2$ et $M = 10^4$ nous obtenons respectivement $p \approx 1.007$ et $p \approx 1.013$. Cette limite n'est pas restrictive en pratique, car les expériences montrent que l'on peut bénéficier des avantages numériques d'un paramètre p proche de 1 sans que p ne descende en dessous d'environ 1.1. Si malgré tout cette borne est dérangeante, une possibilité théorique, quoique potentiellement difficile à implémenter en pratique, serait de normaliser les variables du problème pour avoir $M \approx 1$.

Enfin, le terme $J'_q(H_k(x^*_k))$ dans la recherche de pas n'implique pas de contrainte sur p plus forte. En effet, pour tout vecteur u, $J'_q(u^*) = J'_q(J_p(u)) = (q-1)\underset{i}{\text{diag}}(|u^i|^{(p-1)(q-2)}) = (q-1)\underset{i}{\text{diag}}(|u^i|^{2-p})$.

L'algorithme

En résumé l'algorithme s'écrit

Algorithme 7 Algorithme de la descente de gradient dans le dual avec recherche de pas linéaire

1: Choisir x_0 (tester éventuellement $x_0 = 0$ à part si mise en place de la stratégie de garde-fou)
2: $p_0 \leftarrow -\nabla f_0$
3: $x^*_0 \leftarrow J_p(x_0)$
4: **pour** $k = 0..N-1$ **faire**
5: Calcul du pas α_k satisfaisant soit (4.9), soit (4.10) et (4.11)
6: $x^*_{k+1} \leftarrow x^*_k + \alpha_k p_k$
7: $x_{k+1} \leftarrow J_q(x^*_{k+1})$
8: $p_{k+1} \leftarrow -\nabla f_{k+1}$
9: **fin pour**
10: $x_N \leftarrow J_q(x^*_N)$

L'algorithme est ici écrit pour une application en assimilation de données avec une boucle **for** et un nombre fixé d'itérations, qui peut être remplacée par d'autres conditions d'arrêts.

4.2.3 Gradient conjugué non linéaire dans l'espace dual

Nous adaptons à présent l'algorithme du gradient conjugué non linéaire pour effectuer les itérations dans l'espace dual. La différence avec l'algorithme vu précédemment se situe

entre les équations (4.5) et (4.17) : maintenant p_{k+1} est combinaison linéaire de $-\nabla f_{k+1}$ et de la direction de descente précédente p_k.

$$p_0 = -\nabla f_0, \tag{4.14}$$
$$x^*_{k+1} = x^*_k + \alpha_k p_k, \tag{4.15}$$
$$x_{k+1} = J_p(x^*_{k+1}), \tag{4.16}$$
$$p_{k+1} = -\nabla f_{k+1} + \beta_k p_k. \tag{4.17}$$

Pour $p = q = 2$ on retrouve l'algorithme de gradient conjugué non linéaire classique. Le pas α_k de l'étape (4.15) peut à nouveau être calculé de manière à respecter les conditions de Wolfe ou d'Armijo sur $f \circ J_q$. Il reste à préciser le choix du paramètre β_k.

Choix de β_k

De multiples choix sont possibles et dépendent du choix d'une heuristique. Nous pouvons toujours donner à β_k les valeurs classiques de Fletcher-Rieves (2.9) ou d'Hestenes-Stiefel (2.10). Une première heuristique plus intuitive dans notre cas est de reprendre le paramètre de Fletcher-Rieves et de remplacer simplement f par $f \circ J_q$ ce qui donne

$$\beta_k^{FR,dual} = \frac{\nabla(f \circ J_q)(x^*_{k+1})^T \nabla(f \circ J_q)(x^*_{k+1})}{\nabla(f \circ J_q)(x^*_k)^T \nabla(f \circ J_q)(x^*_k)}. \tag{4.18}$$

Nous pouvons également reprendre le raisonnement pris par Hestenes et Stiefel (voir section (2.1.7)) et demander cette fois à deux directions consécutives d'être conjuguées par rapport à la hessienne moyenne de $f \circ J_q$ sur l'intervalle $[x^*_k, x^*_{k+1}]$:

$$G_k = \int_0^1 \nabla^2 (f \circ J_q)(x^*_k + \tau \alpha_k p_k)\, d\tau.$$

Par un développement de Taylor on a

$$\nabla(f \circ J_q)(x^*_{k+1}) = (f \circ J_q)(x^*_k) + \alpha_k G_k p_k$$
$$G_k p_k = \frac{1}{\alpha_k}\left(\nabla(f \circ J_q)(x^*_{k+1}) - (f \circ J_q)(x^*_k)\right)$$

Imposer la condition $p_{k+1}^T G_k p_k = 0$ fournit :

$$\frac{1}{\alpha_k}(-\nabla f_{k+1} + \beta_k^{HS,dual} p_k)^T (\nabla(f \circ J_q)(x^*_{k+1}) - \nabla(f \circ J_q)(x^*_k)) = 0.$$

En écrivant $(f \circ J_q)(x^*_{k+1}) = (f \circ J_q)_{(k+1)^*}$ on a donc

$$\beta_k^{HS,dual} = \frac{\nabla f_{k+1}^T(\nabla(f \circ J_q)_{(k+1)^*} - \nabla(f \circ J_q)_{k^*})}{p_k^T(\nabla(f \circ J_q)_{(k+1)^*} - \nabla(f \circ J_q)_{k^*})}.$$

ou, de manière plus concise en notant $y_k^* = \nabla(f \circ J_q)_{(k+1)^*} - \nabla(f \circ J_q)_{k^*}$,

$$\beta_k^{HS,dual} = \frac{\nabla f_{k+1}^T y_k^*}{p_k^T y_k^*}. \tag{4.19}$$

Ces choix de β_k ne garantissent pas à p_{k+1} d'être une direction de descente pour f en x_{k+1} ou pour $f \circ J_q$ en x_{k+1}^*. Pour circonvenir à cette éventualité, nous pouvons faire du backtracking sur β_k jusqu'à obtenir une direction de descente pour f_{k+1} et $(f \circ J_q)_{(k+1)^*}$, ce qui sera éventuellement le cas puisque $\lim_{\beta_k \to 0} p_k = -\nabla f_k$.

L'algorithme

Nous verrons qu'en théorie un restart périodique de l'algorithme (poser $\beta_k = 0$ toutes les $K \in \mathbb{N}^*$ itérations) permet d'obtenir la convergence. Nous reviendrons sur la nécessité d'un tel restart à la section (4.4.2) mais donnons l'algorithme le plus général possible en gardant cette étape.

Algorithme 8 Algorithme du gradient conjugué non linéaire avec itérations dans le dual et recherche du pas linéaire

1: Choisir c dans $(0;1)$, x_0, et $n^{\beta}_{itermax}$, $K \in \mathbb{N}^*$
2: $x_0^* \leftarrow J_p(x_0)$
3: $p_0 \leftarrow -\nabla f_0$
4: **pour** $k = 0..N-1$ **faire**
5: Calcul du pas α_k vérifiant soit (4.9), soit (4.10) et (4.11)
6: $x^*_{k+1} \leftarrow x^*_k + \alpha_k p_k$
7: **si** k est un multiple de K **alors**
8: $\beta_k = 0$
9: **sinon**
10: $\beta_k \leftarrow \frac{\nabla f_{k+1}^T y_k^*}{p_k^T y_k^*}$
11: $l \leftarrow 0$
12: **tant que** $\langle (f \circ J_q)'(x_k^*), -\nabla f_{k+1} + \beta_k p_k \rangle > 0$ and $l < n^{\beta}_{\text{itermax}}$ **faire**
13: $\beta_k \leftarrow c\beta_k$
14: $l \leftarrow l + 1$
15: **fin tant que**
16: **si** $l = n^{\beta}_{itermax}$ **alors**
17: $\beta_k \leftarrow 0$
18: **fin si**
19: $p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_k p_k$
20: **fin si**
21: **fin pour**
22: $x_N \leftarrow J_q(x_N^*)$

Les étapes des lignes 12 à 15 concernent le backtracking sur β_k pour s'assurer que p_{k+1} est une direction de descente pour f et $f \circ J_q$. Nous nous donnons un nombre maximal d'itérations possibles $n^{\beta}_{itermax}$ pour restarter l'algorithme si le backtracking est trop long (étapes 16 à 18).

De la même manière que pour l'algorithme (7) nous nous donnons un nombre fixé d'itérations. Plus généralement la boucle **for** peut être remplacée par n'importe quelle condition d'arrêt.

4.2.4 Gradient conjugué non linéaire avec transport de la direction dans le primal

De la même manière que nous avons confectionné l'algorithme du gradient conjugué non linéaire dans l'espace dual à partir de l'algorithme de descente de gradient dans l'espace dual, on peut penser à adapter l'algorithme du gradient conjugué non linéaire à partir de l'algorithme de descente de gradient avec transport du gradient dans l'espace primal. Le changement s'effectuera simplement pour la direction p_{k+1} qui sera construite à partir de $-J_q(\nabla f_{k+1})$ et p_k. Le schéma itératif est donc le suivant :

$$\begin{aligned} p_0 &= -J_q(\nabla f_0), \\ x_{k+1} &= x_k + \alpha_k p_k, \\ p_{k+1} &= J_q(-\nabla f_{k+1} + \beta_k J_p(p_k)). \end{aligned} \tag{4.20}$$

Nous aurions pu calculer p_{k+1} comme la somme de deux éléments ramenés séparément dans l'espace primal, c'est à dire à remplacer (4.20) par $p_{k+1} = J_q(-\nabla f_{k+1}) + \beta_k p_k$. Il semble néanmoins plus avisé d'effectuer la mise à jour de p_{k+1} entièrement dans le dual au lieu de sommer deux éléments qui n'ont a priori plus de liens à cause de la non linéarité de J_q. Cette piste moins intuitive n'est donc pas explorée ici.

Choix de β_k

Sous cette forme, cet algorithme est similaire à l'algorithme du gradient conjugué non linéaire classique à l'exception de la mise à jour de la direction. Ainsi, l'heuristique du choix de β_k est la même que pour celle du gradient conjugué non linéaire. Nous pouvons donc garder les valeurs proposées par Fletcher-Rieves (2.9) et Hestenes-Stiefel (2.10). Seul le paramètre de Hestenes-Stiefel fait intervenir la direction p_k et aura donc des valeurs différentes du cas classique.

Sensibilité numérique par rapport au choix de p

Par rapport à la descente de gradient avec transport des itérés dans le dual, ce type de descente utilise l'opérateur de dualité sur la direction de descente et non plus sur les itérés. Lorsque $p_k = -\nabla f_k$, la sensibilité numérique de l'algorithme dépendra de l'amplitude du gradient et du paramètre q (qui augmente d'autant plus que p se rapproche de 1). Raisonnons sur une seule composante de ∇f_k. Notons M un des ses majorants. Numériquement il faut $J_q(M) \leq N$ où encore $M^{q-1} \leq N$, soit :

$$q \leq \frac{\log(N)}{\log(M)} + 1.$$

Transcrite sur p cette condition donne

$$p \geq \frac{\log M}{\log N} + 1.$$

Inversement, à p donné, la valeur maximale des composantes du gradient pour éviter un débordement en virgule flottante ne doit pas dépasser

$$M \leq N^{p-1}.$$

Nous retrouvons la même limite numérique pour p que pour l'algorithme précédent, mais cette fois M désigne un majorant des composantes du gradient. Cette valeur peut s'avérer bien plus grande que les composantes de x notamment pour les premières itérations.

Au-delà de cette limite numérique, nous nous apercevrons sur les tests numériques en petite dimension que la norme de la direction $J_q(\nabla f_k)$ peut devenir très élevé et peut, sans nécessairement causer d'erreurs numériques, nécessiter des valeurs d'hyperparamètres pour le calcul du pas plus difficile à évaluer.

L'algorithme

Les itérés étant dans l'espace primal, nous demandons au pas α_k de vérifier les conditions de Wolfe classique (2.3) et (2.4). Nous effectuerons toujours pour β_k un backtracking pour s'assurer d'avoir une direction de descente avant un nombre raisonnable d'itérations $n^{\beta}_{itermax}$. De la même manière que pour l'algorithme du gradient conjugué non linéaire dans l'espace dual (ou l'algorithme du gradient conjugué non linéaire classique), nous restartons l'algorithme toutes les $K \in \mathbb{N}^*$ itérations ce qui permet d'assurer la convergence théorique.

Algorithme 9 Algorithme du gradient conjugué non linéaire avec transport de la direction dans l'espace primal

Choisir c dans $(0;1)$, x_0, et $n^{\beta}_{itermax}$, $K \in \mathbb{N}^*$
$p_0 \leftarrow -J_q(\nabla f_0)$
pour $k = 0..N-1$ **faire**
    Calcul du pas α_k vérifiant soit (2.3), soit (2.3) et (2.4) simultanément
    $x_{k+1} \leftarrow x_k + \alpha_k p_k$
    si k est un multiple de K **alors**
        $\beta_k = 0$
    sinon
        Calcul de β_k
        $l \leftarrow 0$
        tant que $\langle f'(x_k), J_q(-\nabla f_{k+1} + \beta_k J_p(p_k))\rangle > 0$ and $l < n^{\beta}_{\text{itermax}}$ **faire**
            $\beta_k \leftarrow c\beta_k$
            $l \leftarrow l + 1$
        fin tant que
        si $l = n^{\beta}_{itermax}$ **alors**
            $\beta_k \leftarrow 0$
        fin si
        $p_{k+1} \leftarrow -J_q(\nabla f_{k+1} + \beta_k p_k)$
    fin si
fin pour

4.3 Comparaisons expérimentales des algorithmes

L'objectif de cette section est de tester en pratique les algorithmes proposés afin de pouvoir les comparer et savoir dans quelles situations lesquels sont les plus adéquats.

Nous testons ces différentes méthodes toujours dans le cas du problème d'advection linéaire. La Figure (3.5) du chapitre précédent, montrant la RMSE et la MAE obtenues au terme d'une descente de gradient pour plusieurs expériences d'assimilation du schéma d'advection, a montré la robustesse de l'algorithme vis-à-vis du problème : les résultats changent peu d'une expérience à l'autre. En particulier il n'y a pas de valeur aberrante. Nous allons donc faire varier λ et p puis minimiser Ω_p avec les différents algorithmes. Les données présentées seront alors la moyenne, pour chaque couple (λ, p) fixé, des résultats obtenus sur 10 expériences.

Nous gardons le même critère de convergence qu'au chapitre précédent, à savoir

$$\|\nabla\Omega_p(\mathbf{x}_k, \mathbf{b}, \lambda)\|_2 < 10^{-4}(\|\nabla\Omega_p(\mathbf{x}_0, \mathbf{b}, \lambda)\|_2 + \sqrt{\varepsilon_{machine}}).$$

Deux autres conditions d'arrêt sont toujours présentes : nous imposons un nombre d'itérations maximal d'une valeur $nb_{it}^{max} = 10^5$, et un critère de stagnation des itérés $\|x_{k+1} - x_k\|_2 \leq 10^{-12}(\|x_k\|_2 + \sqrt{\varepsilon})$.

Sauf indication contraire, les valeurs des constantes pour la recherche de pas sont $c_1 = 10^{-3}$ pour le coefficient du critère d'Armijo, $\rho = \frac{1}{2}$ pour le ratio de backtracking sur le pas, i.e. $\alpha_k^{n+1} = \rho\alpha_k^n$ et le pas initial est pris égal à $\alpha_k^0 = 1$.

Avant de s'intéresser aux vitesses de convergence des algorithmes, nous allons légitimer expérimentalement les choix du paramètre β_k proposés dans les sections précédentes et choisir le plus pertinent pour effectuer les comparaisons.

4.3.1 Comparaison par rapport au choix de β_k

Cette sous-section n'a pas pour but une étude précise de la dépendance entre les algorithmes et les valeurs de β_k proposées. Nous souhaitons simplement montrer la pertinence des paramètres proposés sur un exemple tout en s'assurant d'une robustesse relative par rapport aux valeurs de p et λ. Nous testons donc la valeur $\beta_k = 0$ pour tout k puis les valeurs de Fletcher-Rieves β_k^{FR} (2.9) et Hestenes-Stiefel β_k^{HS} (2.10), ainsi que leur version duale $\beta_k^{FR,dual}$ (4.19) et $\beta_k^{HS,dual}$ (4.18), sur les trois algorithmes de gradient conjugué non linéaire : le classique, celui avec les itérés dans le dual et celui avec transport de la direction dans le primal. Le nombre d'itérations avant convergence pour p appartenant à $\{1.1; 1.5; 2\}$ et pour λ appartenant à $\{0; 10; 100\}$ est reporté sur les tables (4.1) à (4.5).

0. Les abréviations utilisées sont listées ici : **PCG** : Preconditioned Conjugate Gradient / Gradient conjugué préconditionné ; **NLCG** : Non-Linear Conjugate Gradient / Gradient conjugué non linéaire ; **NLCGDS** : Non-Linear Conjugate Gradient Dual Space / Gradient conjugué non linéaire dans l'espace dual ; **GDDS** : Gradient Descent Dual Space / Descente de gradient dans l'espace dual.

Commençons par deux remarques attendues : pour $\lambda = 0$, et uniquement pour NLCG, le nombre d'itérations ne varie en fonction de p. De même, comme espéré, nous observons une amélioration de la vitesse de convergence pour le gradient conjugué non linéaire classique entre le choix $\beta_k = 0$ et les choix $\beta_k = \beta_k^{HS}$ ou β_k^{FR}, le paramètre de Hestenes-Stiefel remportant même la compétition ici. Remarquons alors que, de la même manière, les paramètres duaux $\beta_k^{FR,dual}$ et $\beta_k^{HS,dual}$ permettent de faire converger les deux nouveaux algorithmes de gradients conjugués non linéaire plus vite. Les meilleurs résultats en général sont obtenus par le gradient conjugué non linéaire dans le dual avec ces deux nouvelles valeurs de β_k.

Notons que pour $p = 1.1$, quel que soit λ et quelle que soit la valeur de β_k, l'algorithme NLCG avec transport de la direction dans le primal diverge (noté Div. dans les tableaux), i.e. $\|x_k\|_2 \to \infty$. En fait, cette valeur de p est trop proche de 1 et fait exploser numériquement la norme de p_k dès les premières itérations.

Certaines valeurs de β_k semblent fonctionner pour des algorithmes pour lesquels elles n'étaient pas conçues initialement. Ainsi β_k^{HS} et β_k^{FR} permettent aux deux nouveaux algorithmes de gradient conjugués non linéaire de converger (moins bien que leur contrepartie duale lorsque p s'approche de 1). Pareillement, $\beta_k^{FR,dual}$ et $\beta_k^{HS,dual}$ permettent au NLCG standard de converger, mais moins vite que pour leur contrepartie classique. C'est en fait la stratégie de backtracking sur ce paramètre qui nous permet toujours de retrouver une direction de descente, mais puisque β_k n'a pas de raison d'être pertinent pour la construction de p_{k+1} dans ce scénario, on se ramène à faire des itérations de descente de gradient qui viennent ralentir la convergence.

TABLE 4.1 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour $\boldsymbol{\beta}_{\mathbf{k}}^{\mathbf{HS}}$.

(a) NLCG

λ \ p	1.1	1.5	2
0	44	44	44
10	65	46	72
100	60	48	59

(b) NLCGDS-1

λ \ p	1.1	1.5	2
0	3867	68	48
10	2022	70	48
100	634	82	76

(c) NLCGDS-2

λ \ p	1.1	1.5	2
0	Div.	221	48
10	Div.	247	48
100	Div.	196	76

TABLE 4.2 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour $\boldsymbol{\beta}_{\mathbf{k}}^{\mathbf{FR}}$.

(a) NLCG

λ \ p	1.1	1.5	2
0	177	177	177
10	78	127	92
100	127	158	69

(b) NLCGDS-1

λ \ p	1.1	1.5	2
0	103	70	59
10	159	53	47
100	443	54	63

(c) NLCGDS-2

λ \ p	1.1	1.5	2
0	Div.	101	57
10	Div.	75	101
100	Div.	84	98

TABLE 4.3 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour $\boldsymbol{\beta}_{\mathbf{k,dual}}^{\mathbf{HS}}$.

(a) NLCG

λ \ p	1.1	1.5	2
0	218	61	44
10	162	126	72
100	513	70	59

(b) NLCGDS-1

λ \ p	1.1	1.5	2
0	91	66	48
10	115	50	48
100	106	41	76

(c) NLCGDS-2

λ \ p	1.1	1.5	2
0	Div.	74	68
10	Div.	78	69
100	Div.	65	73

TABLE 4.4 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour $\boldsymbol{\beta}_{\mathbf{k,dual}}^{\mathbf{FR}}$.

(a) NLCG

λ \ p	1.1	1.5	2
0	546	922	177
10	232	394	92
100	3079	211	69

(b) NLCGDS-1

λ \ p	1.1	1.5	2
0	136	55	59
10	197	61	47
100	183	58	63

(c) NLCGDS-2

λ \ p	1.1	1.5	2
0	Div.	65	57
10	Div.	100	101
100	Div.	125	98

TABLE 4.5 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour $\boldsymbol{\beta}_{\mathbf{k}} = \mathbf{0}$.

(a) NLCG

λ \ p	1.1	1.5	2
0	252	252	252
10	250	257	268
100	322	248	248

(b) NLCGDS-1

λ \ p	1.1	1.5	2
0	754	258	252
10	782	171	268
100	1054	309	248

(c) NLCGDS-2

λ \ p	1.1	1.5	2
0	Div.	242	252
10	Div.	241	268
100	Div.	221	248

4.3.2 Comparaison des vitesses de convergence

Nous utilisons à présent exclusivement la valeur β_k^{HS} pour NLCG et la valeur $\beta_k^{HS,dual}$ pour NLCGDS pour pouvoir comparer les vitesses de convergence des différents algorithmes toujours avec les mêmes critères d'arrêt. Nous faisons ici la différence entre NLCGDS-1 qui est l'algorithme de gradient conjugué non linéaire dans le dual **avec recherche de pas dans le dual** et NLCGDS-2 qui correspond au même algorithme **mais avec transport de la direction dans le primal**. Commençons par regarder le nombre d'itérations nécessaire à la convergence en fonction de p et λ. Ces données sont regroupées dans les Tables (4.6) pour les algorithmes de type descente de gradient et (4.7) pour les algorithmes de type gradient conjugué non linéaire.

D'abord en ce qui concerne la Table (4.6), globalement, les meilleurs résultats sont obtenus pour la descente de gradient avec transport de la direction dans le primal. Pour $p = 2$ cet algorithme est bien identique à la descente de gradient classique. Puis, quel que soit λ, quand p décroît jusqu'à 1.6, ces deux algorithmes obtiennent sensiblement les mêmes résultats. Mais lorsque p se rapproche de 1 la descente de gradient classique devient la plus lente des deux. Lorsque $p = 1.1$, le nombre d'itérations requis pour converger augmente exponentiellement en fonction de λ, dépassant les 10^4 pour λ supérieur à 50 et se heurtant au nombre maximal d'itérations allouées à partir de $\lambda = 70$. Cette augmentation est due à la norme de la direction $-J_q(\nabla f_k)$ qui augmente brutalement pour cette valeur de p : la limite inférieure de p évoquée dans la présentation de l'algorithme se manifeste déjà pour cet exemple de faible dimension.

Alors que la descente de gradient dans le dual est globalement la moins efficace des trois méthodes, et ce d'autant plus que λ tend vers 0 et p se rapproche de 2, elle permet tout de même de converger rapidement pour $p = 1.1$ et ce pour tout λ. Pour toutes les valeurs de p, son efficacité augmente même lorsque λ augmente, c'est à dire que le poids de la régularisation en norme L_p augmente. Cet algorithme apparaît alors intéressant pour des expériences nécessitant une forte régularisation. Autrement, les deux autres choix s'avèrent plus avantageux.

Quant à la table (4.7), les résultats qui s'y trouvent présentent d'abord une nette amélioration par rapport aux techniques de descente de gradient précédentes. La troisième table correspondant au gradient conjugué non linéaire avec transport de la direction dans le primal souffre cependant de plusieurs écueils : à nouveau, une valeur de p trop proche de 1 mène à des directions de descente aux normes aberrantes et à une divergence des itérés. De plus, presque aléatoirement dans le tableau, le nombre d'itérations passe brusquement de l'ordre de la centaine à l'ordre de la dizaine de milliers. Malgré ce nombre très élevé, les itérés se sont rapprochés rapidement de la solution du problème mais la norme des directions très élevée (due à l'utilisation de l'opérateur J_q) implique des pas très petits. Les itérés ont alors du mal à se rapprocher du minimum à la précision demandée. Ainsi l'algorithme converge bien en un nombre petit d'itérations (inférieur à 100) si la précision du critère d'arrêt diminue de 10^{-4} à 10^{-3}. Une autre manière de faire converger l'algorithme en un nombre raisonnable d'itérations et de modifier le paramètre c de la recherche de pas. Ainsi en passant de $c = 10^{-3}$ à 10^{-5} nous autorisons des pas plus grands. Cependant en relâchant ce critère nous augmentons globalement le nombre d'itérations requis pour la convergence.

Le second tableau du gradient conjugué non linéaire dans le dual obtient cette fois les meilleurs résultats globaux, malgré les moins bonnes performances de l'algorithme de descente de gradient dans le dual (vis-à-vis des autres descentes de gradient) sur lequel il est construit. Cette méthode est robuste par rapport aux choix de p et λ et apparaît comme le choix le plus judicieux dans ce contexte.

TABLE 4.6 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour les algorithmes de descente de gradient.

(a) Descente de gradient classique

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	1552	1552	1552	1552	1552	1552	1552	1552	1552	1552
10	1345	1161	1187	1235	1281	1318	1358	1385	1406	1432
20	1731	1111	1070	1077	1123	1169	1226	1260	1299	1340
30	2557	1177	970	1031	1018	1063	1112	1165	1215	1257
40	4746	1168	932	960	948	982	1033	1089	1139	1186
50	17837	1200	906	873	917	916	965	1019	1079	1125
60	50531	1195	877	825	898	870	908	966	1016	1078
70	63058	1258	859	793	839	847	863	910	965	1021
80	81654	1281	833	759	810	817	833	867	923	986
90	89160	1321	828	730	774	794	799	834	883	941
100	99560	1412	795	706	725	776	778	808	855	905

(b) Descente de gradient dans l'espace dual

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	11076	7913	2685	1702	3630	4068	5092	6270	2265	5541
10	5854	4975	1826	1171	2094	4073	4604	5523	2458	5115
20	4642	4045	1348	1170	2224	3706	4707	5816	2163	4538
30	4008	3519	1276	1123	1876	3101	5099	4891	2059	4883
40	3063	2920	1063	905	1825	3417	3997	4475	2041	4419
50	2705	2755	1063	1013	1835	2542	3638	4393	1967	4389
60	2643	2452	983	930	1218	2471	3632	4242	1786	4142
70	2700	2268	1050	926	1258	2531	3721	3948	1683	4105
80	2621	2140	944	942	1281	2220	3395	3798	1589	3810
90	2404	2073	821	892	961	1993	3008	3749	1297	3775
100	2376	1958	828	775	1119	1816	2860	3630	1452	3846

(c) Descente de gradient avec transport de la direction dans le primal

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	688	927	1132	1258	1339	1544	1588	1503	2198	1552
10	440	631	808	956	1084	1286	1362	1359	1953	1432
20	453	573	690	816	943	1132	1209	1226	1749	1340
30	4621	534	621	744	834	994	1119	1132	1625	1257
40	65739	522	582	670	762	890	1011	1064	1511	1186
50	92471	517	550	640	699	852	944	1004	1406	1125
60	93405	513	530	595	669	797	869	943	1328	1078
70	nb_{it}^{max}	511	498	566	635	753	811	895	1250	1021
80	nb_{it}^{max}	577	478	536	602	708	781	852	1200	986
90	nb_{it}^{max}	636	458	516	585	687	787	813	1130	941
100	nb_{it}^{max}	952	439	501	549	665	776	790	1095	905

TABLE 4.7 – Nombre d'itérations pour converger en fonction de p et λ (moyenne sur 10 expériences) pour les algorithmes de type gradient conjugué non linéaire.

(a) Gradient conjugué non linéaire classique

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	100	100	100	100	100	100	100	100	100	100
10	116	102	91	93	104	101	86	114	109	133
20	101	121	111	128	110	113	117	96	93	107
30	101	147	121	99	89	91	104	110	106	108
40	130	131	116	106	116	112	116	101	103	106
50	105	113	116	139	103	101	102	102	102	82
60	100	101	141	117	115	115	106	113	89	96
70	96	92	94	106	100	106	108	105	102	100
80	102	101	113	99	104	97	83	117	113	121
90	138	101	93	109	96	100	110	125	111	125
100	142	109	139	93	114	94	91	106	104	123

(b) Gradient conjugué non linéaire dans le dual

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	129	77	80	67	65	71	58	64	69	71
10	130	90	75	67	65	65	57	63	71	73
20	131	89	74	65	65	65	54	63	68	62
30	130	90	71	65	64	68	55	65	66	64
40	128	83	78	61	63	67	58	67	66	66
50	134	88	74	65	63	64	56	63	65	64
60	124	95	79	64	64	69	56	64	60	63
70	126	88	74	70	61	63	58	65	57	54
80	121	89	75	62	61	61	56	64	59	55
90	130	89	72	59	61	65	54	64	54	56
100	130	90	75	60	61	58	55	65	56	56

(c) Gradient conjugué non linéaire avec transport de la direction dans le primal

λ \ p	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
0	Div.	118	122	117	89	10094	88	91	10080	92
10	Div.	127	102	95	87	10079	98	99	75	88
20	Div.	109	10111	93	96	92	10091	101	77	10071
30	Div.	111	10099	115	101	20072	90	76	79	89
40	Div.	130	120	90	107	87	104	10081	82	90
50	Div.	143	112	106	97	120	10090	10085	102	113
60	Div.	126	10102	95	97	95	88	104	91	78
70	Div.	120	138	102	96	95	10090	88	86	86
80	Div.	124	121	129	96	95	101	85	89	93
90	Div.	120	124	120	10099	98	96	92	75	10069
100	Div.	121	6445	87	93	94	98	124	93	89

4.4 Convergence des algorithmes

La preuve de convergence des algorithmes proposés se basent sur les mêmes hypothèses qui servent au théorème de Zoutendijk classique (2.1.2) rappelé au chapitre un. À l'exception d'une seule différence : les normes intervenant dans l'hypothèse du gradient lipschitzien dépendent à présent des espaces de départ et d'arrivée de f et ne sont plus nécessairement des normes euclidiennes.

Hypothèse 4.4.1. — *(i) L'ensemble de niveau $\mathcal{L} = \{x : f(x) \leq f(x_0)\}$, où x_0 est le point de départ des itérations, est borné.*

— *(ii) La fonction objectif f est continûment différentiable sur un voisinage ouvert $\mathcal{N}$ de $\mathcal{L}$, et son gradient est lipschitzien : il existe $L > 0$ tel que*

$$\forall x, \tilde{x} \in N, \|\nabla f(x) - \nabla f(\tilde{x})\| \leq L\|x - \tilde{x}\|.$$

Nous spécifions les normes intervenant dans l'hypothèse *ii* lorsque nous l'utiliserons. Comme remarqué dans ([67]) ces hypothèses impliquent l'existence d'une constante $\bar{\gamma_1} > 0$ telle que

$$\forall x \in N, \|\nabla f(x)\| \leq \bar{\gamma_1}.$$

Étant donné l'expression de J'_q (4.6), elles impliquent également l'existence d'une constante $\bar{\gamma_2} > 0$ telle que

$$\forall x \in N, \|J'_q(J_p(x))\nabla f(x)\| \leq \bar{\gamma_2}.$$

Nous commençons par montrer la convergence de l'algorithme de descente de gradient avec transport des itérés dans l'espace dual lorsque la recherche de pas est soumise aux conditions de Wolfe. Ces résultats permettront d'obtenir la convergence de l'algorithme du gradient conjugué non linéaire avec recherche de pas dans l'espace dual proposé, lorsqu'une stratégie de backtracking sur le pas α est utilisée.

4.4.1 Algorithme de descente de gradient dans l'espace dual

Pour montrer la convergence de cet algorithme de descente, nous nous inspirons du théorème de Zoutendijk (2.1.2) que nous adaptons pour des algorithmes avec itérations et recherche de pas dans l'espace dual.

Théorème 4.4.1 (Théorème de Zoutendijk pour des conditions de Wolfe dans l'espace dual)**.** *Soit un schéma itératif de la forme $x^*_{k+1} = x^*_k + \alpha_k p_k$ où p_k est une direction de descente pour f_k et pour $(f \circ J_q)(x^*_k)$, et $\nabla f_k^T J'_q(\nabla f_k)p_k < 0$. Soit un pas α_k satisfaisant (4.10) and (4.11). Supposons l'hypothèse (4.4.1). L'hypothèse d'un gradient de f lipschitzien s'écrit ici*

$$\forall x, \tilde{x} \in N, \|\nabla f(x) - \nabla f(\tilde{x})\|_{op} \leq L\|x - \tilde{x}\|_p.$$

avec $\|\nabla f(x)\|_{op}$ *la norme d'opérateur de* $\nabla f(x)$*, vu comme un élément de* $\mathcal{L}((\mathbb{R}^n, \|\cdot\|_p), \mathbb{R})$ [1]*. Alors, en notant*

$$\cos(\theta_k) = \frac{-\nabla f_k^T J_q'(H_k(x_k^*))p_k}{\|\nabla f_k\|_q \|J_q'(H_k(x_k^*))p_k\|_p},$$

il vient

$$\sum_{k\geq 0} \frac{\|\nabla f_k\|_q^q \|J_q'(H_k(x_k^*))p_k\|_p}{\|p_k\|_q} \cos(\theta_k)^q < \infty. \tag{4.21}$$

Remarque 5. *Pour* $p = 2$ *nous retrouvons le théorème de Zoutendijk.*

Preuve 4.4.2 (Preuve du théorème 4.4.1)**.**

L'inégalité (4.11) fournit

$$\nabla f_{k+1}^T J_q'(H_k(x_k^*))p_k - \nabla f_k^T J_q'(H_k(x_k^*))p_k \geq (c_2 - 1)\nabla f_k^T J_q'(H_k(x_k^*))p_k. \tag{4.22}$$

D'après la définition de la norme d'opérateur et grâce à la condition de Lipschitz sur le gradient on a successivement :

$$\begin{aligned}(\nabla f_{k+1} - \nabla f_k)^T (J_q'(H_k(x_k^*))p_k &\leq \|\nabla f_{k+1} - \nabla f_k\|_{op} \|J_q'(H_k(x_k^*))p_k\|_p \\ &\leq L\|J_q(x_{k+1}^*) - J_q(x_k^*)\|_p \|J_q'(H_k(x_k^*))p_k\|_p. \end{aligned} \tag{4.23}$$

$(\mathbb{R}^n, \|\cdot\|_p)$ *est p-lisse et l'ensemble de niveau* $\mathcal{L}$ *est borné par hypothèse. Grâce au corollaire 2.2.19 :* $\|J_q(x^*) - J_q(y^*)\|_p \leq C\|x^* - y^*\|_q^{p-1}$ *pour un certain* $C > 0$*. Ainsi*

$$(\nabla f_{k+1} - \nabla f_k)^T J_q'(H_k(x_k^*))p_k \leq LC\alpha_k^{p-1}\|p_k\|_q^{p-1}\|J_q'(H_k(x_k^*))p_k\|_p. \tag{4.24}$$

Posons $K = LC$*. Grâce à (4.22) et (4.24) on a :*

$$\alpha_k \geq \left(\frac{1 - c_2}{K} \frac{-\nabla f_k^T J_q'(H_k(x_k^*))p_k}{\|p_k\|_q^{p-1}\|J_q'(H_k(x_k^*))p_k\|_p}\right)^{\frac{1}{p-1}}.$$

d'où

$$c_1\alpha_k \nabla f_k^T J_q'(H_k(x_k^*))p_k \leq c_1\left(\frac{1 - c_2}{K}\right)^{\frac{1}{p-1}} \left(\frac{-\nabla f_k^T (J_q'(H_k(x_k^*))p_k}{\|p_k\|_q^{p-1}\|J_q'(H_k(x_k^*))p_k\|_p}\right)^{\frac{1}{p-1}} \nabla f_k^T J_q'(H_k(x_k^*))p_k$$

1. Par définition, sous ce point de vue, $\|\nabla f(x)\|_{op} = \sup\{|\langle \nabla f(x), h\rangle| \ : \ h \in (\mathbb{R}^n, \|\cdot\|_p) \ \text{ et } \ \|h\|_p \leq 1\}$.

car $\nabla f_k^T J_q'(H_k(x_k^*))p_k < 0$ *par hypothèse. Soit* $c = c_1\left(\frac{1-c_2}{K}\right)^{\frac{1}{p-1}}$, *en utilisant (4.10) on en déduit :*

$$f_{k+1} - f_k \leq -\underbrace{c\left(\frac{-\nabla f_k^T J_q'(H_k(x_k^*))p_k}{\|p_k\|_q^{p-1}\|J_q'(H_k(x_k^*))p_k\|_p}\right)^{\frac{1}{p-1}}}_{>0}\underbrace{(-\nabla f_k^T J_q'(H_k(x_k^*))p_k)}_{>0} < 0.$$

En sommant pour k *allant de* 0 *à* $N \in \mathbb{N}^*$ *:*

$$f_0 - f_{N+1} \geq \sum_{k=0}^{N} c\left(\frac{(-\nabla f_k^T J_q'(H_k(x_k^*))p_k)^p}{\|p_k\|_q^{p-1}\|J_q'(H_k(x_k^*))p_k\|_p}\right)^{\frac{1}{p-1}}$$

En invoquant le caractère borné inférieurement de f, *la série correspondante de termes positifs de la partie droite de l'inégalité précédente est convergente. En écrivant*

$$\cos(\theta_k) = \frac{-\nabla f_k^T J_q'(H_k(x_k^*))p_k}{\|\nabla f_k\|_q\|J_q'(H_k(x_k^*))p_k\|_p}, \tag{4.25}$$

nous obtenons

$$\sum_{k\geq 0}\cos(\theta_k)^{\frac{p}{p-1}}\frac{\|\nabla f_k\|_q^{\frac{p}{p-1}}\|J_q'(H_k(x_k^*))p_k\|_p^{\frac{p}{p-1}}}{\|p_k\|_q\|J_q'(H_k(x_k^*))p_k\|_p^{\frac{1}{p-1}}} < \infty.$$

En se souvenant que $q = \frac{p}{p-1}$, *nous avons le résultat souhaité :*

$$\sum_{k\geq 0}\frac{\|\nabla f_k\|_q^q\|J_q'(H_k(x_k^*))p_k\|_p}{\|p_k\|_q}\cos(\theta_k)^q < \infty.$$

Remarque 6. *le cosinus défini à l'équation (4.25) est bien un élément de* $[-1;1]$ *d'après l'inégalité de Hölder.*

En particulier, la descente de gradient avec recherche de pas dans l'espace dual converge :

Corollaire 4.4.3. *Sous les hypothèses du théorème 4.4.1, prendre* $p_k = -\nabla f_k$ *pour tout* k *implique* $\nabla f_k \to 0$.

Preuve 4.4.4 (Preuve du Corollaire 4.4.3)**.**

La série (4.21) est convergente. En substituant p_k par $-\nabla f_k$ nous obtenons

$$\|\nabla f_k\|_q^{q-1}\|J_q'(H_k(x_k^*))\nabla f_k\|_p \cos(\theta_k)^q \to 0. \tag{4.26}$$

Deux cas se présentent selon la valeur prise par $H_k(x_k^*) = x_k^*$. Nous considérons d'abord le cas où $H_k(x_k^*) = x_k^*$; cela se produit quand $\nabla f_k^T J_q'(x_k^*)\nabla f_k \geq \varepsilon$. Or d'après l'inégalité de Hölder :

$$\nabla f_k^T J_q'(x_k^*)\nabla f_k \leq \|\nabla f_k\|_q \|J_q'(x_k^*)\nabla f_k\|_p.$$

Et par conséquent :

$$\begin{aligned}\|\nabla f_k\|_q^{q-1}\|J_q'(x_k^*)\nabla f_k\|_p &\geq \|\nabla f_k\|_q^{q-2}\nabla f_k^T J_q'(x_k^*)\nabla f_k \\ &\geq \varepsilon\|\nabla f_k\|_q^{q-2}.\end{aligned}$$

L'exposant $q-2$ est positif. Pour avoir $\nabla f_k \to 0$ il suffit de montrer que le cosinus est borné inférieurement par une constante strictement positive. L'hypothèse (4.4.1) implique $\|\nabla f_k\|_q \leq \bar{\gamma_1}$ et $\|J_q'(H_k(x_k^*))\nabla f_k\|_p \leq \bar{\gamma_2}$. Nous avons donc dans ce cas

$$\begin{aligned}\cos(\theta_k) &= \frac{\nabla f_k^T J_q'(x_k^*)\nabla f_k}{\|\nabla f_k\|_q \|J_q'(x_k^*)\nabla f_k\|_p} \\ &\geq \frac{\varepsilon}{\bar{\gamma_1}\bar{\gamma_2}} > 0.\end{aligned}$$

Considérons maintenant le cas où $H_k(x_k^*) = \nabla f_k$.
Écrivons $g_k^i = (\nabla f_k)^i$, nous avons

$$J_q'(\nabla f_k)\nabla f_k = (q-1)\begin{bmatrix}\vdots \\ |g_k^i|^{q-2}g_k^i \\ \vdots\end{bmatrix}$$

par conséquent $\|J_q'(\nabla f_k)\nabla f_k\|_1 = (q-1)\sum_i |g_k^i|^{q-1} = (q-1)\|\nabla f_k\|_{q-1}^{q-1} \leq K_1\|\nabla f_k\|_q^{q-1}$ pour un certain $K_1 > 0$ (grâce à l'équivalence des normes en dimension finie). Donc, pour une nouvelle constante $K_2 > 0$,

$$\begin{aligned}\frac{1}{\|J_q'(\nabla f_k)\nabla f_k\|_p} &\geq \frac{K_2}{\|J_q'(\nabla f_k)\nabla f_k\|_1} \\ &\geq \frac{K_2}{K_1\|\nabla f_k\|_q^{q-1}}\end{aligned} \tag{4.27}$$

On a également :

$$\nabla f_k^T J_q'(\nabla f_k)\nabla f_k = (q-1)\sum_i (g_k^i)^2 |g_k^i|^{q-2} = (q-1)\|\nabla f_k\|_q^q. \tag{4.28}$$

En combinant (4.27) et (4.28), et en appelant $K = (q-1)\frac{K_2}{K_1}$ *:*

$$\begin{aligned}\cos(\theta_k) &= \frac{\nabla f_k^T J_q'(\nabla f_k)\nabla f_k}{\|\nabla f_k\|_q \|J_q'(\nabla f_k)\nabla f_k^T\|_p} \\ &\geq K \frac{\|\nabla f_k\|_q^q}{\|\nabla f_k\|_q \|\nabla f_k\|_q^{q-1}} \\ &= K > 0.\end{aligned}$$

À nouveau, le cosinus est borné inférieurement par une constante strictement positive. Quant au second terme de (4.26), on a

$$\begin{aligned}\|J_q'(\nabla f_k)\nabla f_k\|_p^p &= (q-1)^p \sum_i |g_k^i|^{p(q-1)} \\ &= (q-1)^p \sum_i |g_k^i|^q \\ &= (q-1)^p \|\nabla f_k\|_q^q\end{aligned}$$

D'où $\|J_q'(\nabla f_k)\nabla f_k\|_p = (q-1)\|\nabla f_k\|_q^{\frac{q}{p}} = (q-1)\|\nabla f_k\|_q^{q-1}$ *et donc*

$$\|\nabla f_k\|_q^{q-1}\|J_q'(\nabla f_k)\nabla f_k\|_p = (q-1)\|\nabla f_k\|_q^{2(q-1)}$$

Finalement, dans tous les cas, la condition (4.26) implique bien $\nabla f_k \to 0$.

4.4.2 Algorithme de gradient conjugué non linéaire dans le dual

La convergence de l'algorithme du gradient conjugué non linéaire classique s'obtient en faisant un « restart »(i.e. en posant $\beta_k = 0$) toutes les K itérations (avec $K \in \mathbb{N}^*$ fixé). De même nous obtenons la convergence de l'algorithme du gradient conjugué non linéaire dans l'espace dual avec la même stratégie.

En effet en effectuant un restart toutes les K itérations, nous obtenons une suite infinie $(k_1, k_2, ...)$ telle que

$$\sum_{k=k_1,k_2,...} \frac{\|\nabla f_k\|_{op}^q \|J_q'(H_k(x_k^*))p_k\|_p}{\|p_k\|_q} \cos(\theta_k)^q < \infty.$$

D'après le Corollaire (4.4.3) on a

$$\lim_{j\to\infty} \cos(\theta_{k_j}) = 0$$

ou encore $\liminf_{k\to\infty} \cos(\theta_k)^q = 0$.

Dans le cadre de l'assimilation de données où le nombre d'itérations possible est petit et fixé, nous soutenons qu'un restart est superflu. Pour le peu d'itérations allouées sur un problème réel, il est illusoire d'espérer satisfaire en général les conditions d'arrêts classiques (i.e. faire descendre la norme du gradient sous un ε de l'ordre de 10^{-6} par exemple). De même, le garde-fou de l'algorithme de la descente de gradient dans le dual, mis au point pour converger vers un minimum de f et non de $f \circ J_q$, ne doit pas être nécessaire en pratique. Il apparaît donc profitable de faire décroître $f \circ J_q$ plutôt que f pour s'approcher rapidement d'un minimum au moins lors des premières itérations. Nous n'avons pas de preuve de convergence vers un point critique de f pour un algorithme de descente de gradient dans le dual en ne considérant que la condition d'Armijo, même grâce à l'utilisation d'un garde-fou. À l'aune de ces remarques nous pouvons cependant appliquer l'algorithme avec la condition d'Armijo seule pour faire décroître $f \circ J_q$, quitte à éventuellement utiliser, après plusieurs itérations, un algorithme dont nous savons qu'il converge théoriquement.

Nous proposons donc de ne pas implémenter une stratégie de restart périodique pour cet algorithme. Un restart serait nécessaire si le chemin suivi par les itérés ne bénéficiait plus des informations apportées par les premières directions de descentes suivies. Nous conseillons donc plutôt de reprendre soit la stratégie classique de faire un restart lorsque que deux gradients consécutifs sont loin d'être orthogonaux :

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\|\nabla f_k\| \|\nabla f_{k-1}\|} \geq \nu$$

avec $\nu = 0.1$ typiquement ([30]), soit de considérer plutôt « l'angle » introduit à l'équation 4.25 entre deux gradients consécutifs :

$$\frac{-\nabla f_k^T J_q'(H_k(x_k^*)) \nabla f_{k-1}}{\|\nabla f_k\|_q \|J_q'(H_k(x_k^*)) \nabla f_{k-1}\|_p} \geq \nu.$$

4.4.3 Algorithme du gradient conjugué non linéaire avec transport de la direction dans le primal

Puisque seule la direction change par rapport au gradient conjugué non linéaire classique, nous sommes dans les conditions pour utiliser le théorème de Zoutendijk : le schéma itératif est de la forme $x_{k+1} = x_k + \alpha_k p_k$ et on s'assure que p_k est une direction de descente pour f_k à chaque itération.

Nous avons vu qu'avec du restart périodique il suffisait alors d'avoir un cosinus entre p_k et $-\nabla f_k$ borné inférieurement par $\epsilon > 0$ pour obtenir la convergence de la norme de ∇f_k vers 0. Néanmoins avec $p_k = -J_q(\nabla f_k)$ il est difficile a priori d'obtenir cette borne inférieure pour le cosinus :

$$\cos(\theta_k) = \frac{\langle \nabla f_k, J_q(\nabla f_k)\rangle}{\|\nabla f_k\|_2 \|J_q(\nabla f_k)\|_2}.$$

En changeant de perspective nous pouvons malgré tout obtenir un nouveau théorème de Zoutendijk avec une définition différente du cosinus qui rendra cette minoration instantanée. Ce troisième théorème de Zoutendijk requiert de ne plus voir l'angle $\langle p_k, J_q(\nabla f_k)\rangle$ comme un produit scalaire euclidien mais comme un crochet de dualité. Comme pour le Théorème 4.4.1, nous devons voir ∇f_k comme une application linéaire de L_p ou $(\mathbb{R}^n, \|\cdot\|_p)$ dans $\mathbb{R}$ et l'hypothèse du gradient lipschitzien de f s'écrira à nouveau

$$\forall x, \tilde{x} \in N, \|\nabla f(x) - \nabla f(\tilde{x})\|_{op} \leq L\|x - \tilde{x}\|_p. \tag{4.29}$$

Mais, contrairement au Théorème 4.4.1, les itérés sont dans le primal. La démonstration sera donc fortement similaire à celle du théorème de Zoutendijk.

Théorème 4.4.5 (Théorème de Zoutendijk pour des distances non euclidiennes)**.** *Soit un schéma itératif de la forme $x_{k+1} = x_k + \alpha_k J_q(d_k)$ où $J_q(d_k)$ est une direction de descente pour f en x_k et α_k respecte les deux conditions de Wolfe (2.3) et (2.4). Sous l'hypothèse (4.4.1) et l'hypothèse d'un gradient lipschitzien au sens (4.29), en notant*

$$\cos(\theta_k) = -\frac{\langle \nabla f_k, J_q(d_k)\rangle}{\|\nabla f_k\|_q \|J_q(d_k)\|_p}, \tag{4.30}$$

on a

$$\sum_{k\geq 0} \cos^2(\theta_k)\|\nabla f_k\|_q^2 < \infty.$$

Preuve 4.4.6 (Preuve du théorème 4.4.5)**.**

D'après la définition de la norme d'opérateur et l'hypothèse (4.29) on a

$$\begin{aligned}\langle \nabla f_{k+1} - \nabla f_k, J_q(d_k)\rangle &\leq \|\nabla f_{k+1} - \nabla f_k\|_{op}\|J_q(d_k)\|_p \\ &\leq L\|x_{k+1} - x_k\|_p \|J_q(d_k)\|_p \\ &\leq L\alpha_k \|J_q(d_k)\|_p^2.\end{aligned}$$

D'après la seconde condition de Wolfe (2.4) :

$$\langle \nabla f_{k+1} - \nabla f_k, J_q(d_k)\rangle \geq (c_2 - 1)\langle \nabla f_k, J_q(d_k)\rangle.$$

D'où

$$\alpha_k \geq \frac{c_2 - 1}{L} \frac{\langle \nabla f_k, J_q(d_k) \rangle}{\|J_q(d_k)\|_p^2}$$

puis en utilisant la première condition de Wolfe (2.3) :

$$f_{k+1} \leq f_k - c_1 \frac{1 - c_2}{L} \frac{\langle \nabla f_k, J_q(d_k) \rangle^2}{\|J_q(d_k)\|_p^2}.$$

Soit en utilisant la définition (4.30) et en posant $C = c_1 \frac{1-c_2}{L}$ *:*

$$f_{k+1} \leq f_k - C \cos(\theta_k)^2 \|\nabla f_k\|_q^2. \tag{4.31}$$

On conclut de même en sommant et en invoquant le caractère borné inférieurement de f*.*

Grâce à ce nouveau point de vue nous obtenons une nouvelle démonstration de la convergence de la descente de gradient avec transport des itérés dans le primal puisque maintenant nous avons pour $p_k = J_q(-\nabla f_k) = -J_q(\nabla f_k)$:

$$\cos(\theta_k) = \frac{\langle \nabla f_k, J_q(\nabla f_k) \rangle}{\|\nabla f_k\|_q \|J_q(\nabla f_k)\|_p}.$$

Or d'après la définition de l'opérateur de dualité on a les deux relations suivantes, pour tout x appartenant à L_q ou à $(\mathbb{R}^n, \|\cdot\|_q)$,

$$\begin{aligned} \|J_q(x)\|_p &= \|x\|_q^{q-1} \\ \langle J_q(x), x \rangle_{L_p, L_q} &= \|x\|_q^q. \end{aligned}$$

Et donc

$$\cos(\theta_k) = \frac{\|\nabla f_k\|_q^q}{\|\nabla f_k\|_q \|\nabla f_k\|_q^{q-1}} = 1.$$

Nous obtenons ainsi la convergence de la descente de gradient avec transport de la direction dans le primal et par conséquent la convergence du gradient conjugué non linéaire avec transport de la direction dans le primal et restart périodique.

4.5 Conclusion

Nous avons présenté des algorithmes issus des techniques de minimisation dans les espaces de Banach à dessein de minimiser efficacement la fonctionnelle du 4DVar pénalisée par une norme L_p. L'intuition derrière cette volonté se fonde sur la modélisation du problème régularisé qui suppose que les variables considérées appartiennent à l'espace L_p ou, en dimension finie, à $\mathbb{R}^n$ équipé de la norme L_p.

Deux algorithmes de descente de gradient, respectivement avec transport des itérés dans l'espace dual et transport de la direction de descente dans l'espace primal, donnent lieu à deux nouveaux types d'algorithme de gradient conjugué non linéaire. Le premier, avec transport des itérés dans le dual, s'est démarqué en terme de nombre d'itérations pour converger et de robustesse vis-à-vis du choix de p et λ. Le second, avec transport de la direction dans le primal, a au contraire témoigné d'une moins grande robustesse pour des valeurs de p proche de 1.

Les preuves de convergence des algorithmes de descente proposés reposent sur l'adaptation du théorème de Zoutendijk pour des espaces non euclidiens. En particulier, sur une condition d'angle nouvelle entre le gradient et la direction de descente, faisant appel aux normes L_p et L_q. Elles reposent également sur l'utilisation d'un garde-fou (pour les algorithmes de descente de gradient) et d'une technique de restart (pour les algorithmes de gradient conjugué non linéaires), qui peuvent engendrer des calculs supplémentaires, mais dont la nécessité pratique dans le cadre de l'assimilation de données sera interrogée au chapitre suivant.

Chapitre 5

Vers un problème d'assimilation de données plus réaliste : les équations de Barré de Saint-Venant en deux dimensions

5.1 Présentation du problème . . . 116
5.1.1 Dynamique du système . . . 116
5.1.2 Intégration numérique du système . . . 116
5.1.3 Système d'assimilation (génération des observations, fenêtre d'assimilations) . . . 117
5.1.4 Validation numérique . . . 119
5.2 Minimisation du 4DVar pénalisé . . . 120
5.2.1 Choix de la base $\mathbf{\Phi}$ pour la pénalisation . . . 121
5.2.2 Choix des paramètres λ et p . . . 121
5.2.3 Vitesse de convergence des algorithmes . . . 122
5.2.4 Comportement de NLCGDS . . . 124
5.3 Conclusion . . . 127

Nous nous intéressons à la fois aux performances des algorithmes proposés et à l'efficacité de la régularisation en norme L_p sur un problème d'assimilation de données basé sur les équations de Barré de Saint-Venant. Nous commençons par décrire le système d'assimilation mis en place.

5.1 Présentation du problème

5.1.1 Dynamique du système

Le système considéré est un fluide soumis aux équations de Navier-Stokes amorties en eaux peu profondes (ou équations de Barré de Saint-Venant) très présentes en géophysique et en océanographie. Elles résultent des équations de conservation de la masse, de la quantité de mouvement et de l'énergie pour un fluide ici supposé homogène et incompressible, auxquelles est ajoutée l'hypothèse que les échelles verticales de l'écoulement sont négligeables devant les échelles horizontales ([86]). Elles permettent par exemple de décrire les courants de marée. Les équations s'écrivent en coordonnées cartésiennes

$$\begin{cases} \dfrac{\partial u}{\partial t} + u\dfrac{\partial u}{\partial x} + v\dfrac{\partial u}{\partial y} - fv + g\dfrac{\partial z}{\partial x} = \nu\Delta u, \\ \dfrac{\partial v}{\partial t} + u\dfrac{\partial v}{\partial x} + v\dfrac{\partial v}{\partial y} - fu + g\dfrac{\partial z}{\partial y} = \nu\Delta v, \\ \dfrac{\partial z}{\partial t} + u\dfrac{\partial z}{\partial x} + v\dfrac{\partial z}{\partial y} + z(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}) = \nu\Delta z. \end{cases} \tag{5.1}$$

Les inconnues sont les les composantes de la vitesse horizontale du fluide $u(x, y, t)$ et $v(x, y, t)$, ainsi que $z(x, y, t)$ la hauteur du fluide. L'accélération de la pesanteur g est prise constante égale à 10. Le coefficient de Coriolis est donné par l'approximation du plan Beta : $f = f_0 + \beta Y$ avec $f_0 = 2\Omega\sin(\frac{\pi}{4})$, $\Omega = \frac{2\pi}{86400}$ le facteur de Coriolis, $\beta = 2\frac{\Omega\cos(\frac{\pi}{4})}{R}$ le paramètre de Rossby, $R = \frac{1}{2\pi}4.10^7$ le rayon de la Terre et Y la distance méridionale. Le facteur d'amortissement est égal à $\nu = 10^7\ m^2s^{-1}$. Les frictions aux interfaces entre le fluide et l'air et entre le fluide et le fond marin sont négligées.

5.1.2 Intégration numérique du système

L'expérience se déroule sur une grille rectangulaire de taille $L_x = 32 \times 10^6$ m par $L_y = 8 \times 10^6$ m à une altitude de référence de $z_{ref} = 100$m. Cette grille est divisée en $N_x = N_y = 32$ points horizontalement et verticalement.

En ce qui concerne les conditions aux limites, des conditions de Dirichlet sont imposées selon l'axe x et des conditions aux limites périodiques sont imposées selon l'axe y, i.e. : $u(0, y, t) = u_0(y, t)$, $u(L_x, y, t) = u_{L_x}(y, t)$ et $u(x, 0, t) = u(x, L_y, t)$, et de même pour les variables v et z.

Le système est intégré via un schéma saute-mouton (ou « leapfrog scheme ») avec filtre d'Asselin, souvent utilisé en météorologie ([87], [88]) : en posant $X = (u, v, z)$ l'état et

en notant $\frac{dX}{dt} = F(X)$, on intègre l'état entre l'instant n et l'instant $n+1$ via

$$\tilde{X}^{n+1} = X^{n-1} + 2\Delta t F(\tilde{X}^n) \tag{5.2}$$

$$X^n = \tilde{X}^n + \frac{\nu}{2}(\tilde{X}^{n+1} - 2\tilde{X}^n + X^{n-1}). \tag{5.3}$$

5.1.3 Système d'assimilation (génération des observations, fenêtre d'assimilations)

Fenêtres d'assimilations

Deux fenêtres d'assimilation sont considérées, chacune d'une durée $T = 21600\text{s} = 6\text{h}$ (durée typique des fenêtres utilisées dans les centre météorologiques), et le pas de temps de l'intégration numérique est pris égale $dt = 240\text{s}$.

Conditions initiales

Les conditions initiales des composantes **u** et **v** sont prises dans l'approximation géostrophique, c'est à dire que la force de Coriolis et la force du gradient de pression atmosphérique horizontale sont à l'équilibre. Cette approximation est efficace lorsque la latitude est supérieur à 10° et est invalide à l'équateur. En négligeant la friction et en régime permanent avec une faible courbure, cette approximation s'écrit

$$\mathbf{u}_g = -\frac{g}{f}\frac{\partial \mathbf{z}}{\partial \mathbf{y}}$$
$$\mathbf{v}_g = \frac{g}{f}\frac{\partial \mathbf{z}}{\partial \mathbf{x}}.$$

Concernant la hauteur du fluide, nous considérons une condition initiale avec une structure semblable à la condition initiale de l'expérience d'advection linéaire : il s'agit d'un plateau surélevé et relié à la surface plate du fluide par des pentes linéaires (voir Figure 5.1a). La condition initiale pour la seconde fenêtre d'assimilation est celle de la première fenêtre, soumise au modèle numérique pendant un temps T (voir Figure 5.1b). Notons que si la première condition avec une structure quasi-parcimonieuse dans la base de la dérivée du signal par exemple, il n'en est plus de même pour la seconde condition initiale à cause du modèle numérique qui a lissé le signal. Se pose donc la question d'un choix judicieux d'une base pour la pénalisation.

Générations des observations et matrice de covariance d'erreur des observations

Les observations sont effectuées aléatoirement en temps et en espace sur l'intérieur de la grille. La densité temporelle des observations est prise égale à 0.1 : en moyenne une observation survient tous les 10 pas de temps. La densité spatiale des observations est prise égale à 0.04 : lorsqu'une observation se produit, 4% des points de la grille

sont observés. Un exemple d'observations est donné sur la Figure (5.2) Les observations concernent exclusivement l'altitude z et sont bruitées par un bruit gaussien centré sur le véritable état et d'écart-type $\sigma_{obs} = 1e-4$.

La matrice de covariance d'erreur des observations ne prendra pas en compte les erreurs de corrélation spatiale mais seulement des erreurs de corrélations temporelles. Pour ce faire l'opérateur de corrélation temporel utilisé est une fonction gaussienne appliquée aux observations faites au cours du temps, avec une constante $l_t = 6000$s. Cela revient à appliquer la matrice de terme général $C_{i,j} = e^{-\frac{(|i-j|dt_{obs})^2}{2L_t^2}}$ à un vecteur, avec dt_{obs} l'intervalle de temps entre deux observations et $L = \frac{l_t}{dt}$. En pratique il suffit de calculer la factorisation de Cholesky de cette matrice une fois pour l'appliquer efficacement à plusieurs reprises pendant l'assimilation.

Les constantes de Daley ([89]) l_x et l_y valent selon les axes x et y, $l_x = l_y = 0.8 \times 10^6$. Nous vérifions que les rapports $\frac{l_x}{dx}$ et $\frac{l_x}{dy}$ (dx et dy étant le pas de discrétisation selon les deux axes) sont bien inférieurs à 1 pour obtenir des erreurs de corrélation.

Ébauche et matrice de covariance d'erreur du background

L'ébauche pour la première fenêtre d'assimilation est une gaussienne centrée au centre de la grille. L'ébauche est un signal lisse : cela permet de simuler une ébauche provenant d'une assimilation précédente, qui aurait été lissée par le modèle numérique (visible sur la Figure (5.1c). L'ébauche pour la seconde fenêtre d'assimilation sera également l'état analysé de la première fenêtre, propagée par le modèle jusqu'au temps T, et sera par conséquent elle aussi lissée.

Nous avons souligné à la section (1.2.2) l'importance de la matrice de covariance d'erreur du background $\mathbf{B}$. Nous la modélisons ici par un opérateur de diffusion intégré par un schéma d'Euler implicite ([90]). Ainsi, plus deux variables sont distantes dans l'espace, moins elles interagiront entre elles selon un processus de diffusion. L'opérateur s'écrit

$$R_s = \mathbf{D}^{1/2}\mathbf{N}^{1/2}(\mathbf{I} + \kappa\widetilde{dt}(-\Delta)^2)^{-M}\mathbf{N}^{1/2}\mathbf{D}^{1/2} \tag{5.4}$$

avec

- Δ le Laplacien,
- κ le coefficient de diffusion,
- $\widetilde{dt}$ le pas de temps utilisé pour l'intégration,
- M le nombre de pas de temps,
- $\mathbf{N}$ une matrice diagonale de mise à l'échelle $\mathbf{N} = \text{diag}([\mathbf{I} + \kappa\tilde{dt}(-\Delta)^2)^{-M}]^{-1})$,
- $\mathbf{D}$ une matrice diagonale contenant les variances des variables d'erreur de background, prises ici toutes égales à 1.

5.1.4 Validation numérique

Test numérique du gradient

Une difficulté pratique majeure des méthodes variationnelles est la dérivation du modèle adjoint permettant le calcul du gradient de la fonction coût. Comme son nom l'indique, le modèle adjoint est l'opérateur adjoint du modèle linéaire tangent du modèle physique direct. La mise en pratique de toute méthode variationnelle nécessite donc d'écrire le modèle linéaire tangent, puis son adjoint. Les sources d'erreurs sont donc multiples. S'il existe des outils de différentiation automatique, dans notre cas, afin de bien comprendre en détail le fonctionnement de l'adjoint, nous avons choisi d'écrire directement le modèle adjoint à la main. Une phase de validation du modèle adjoint, et donc du gradient, constitue donc une étape obligatoire avant de pouvoir lancer les différentes optimisations. La méthode choisie pour cela s'appelle le test du gradient.

Un développement de Taylor à l'ordre 1 de la fonction coût Ω autour d'un certain $\widetilde{\mathbf{x}}$ nous donne

$$\Omega(\widetilde{\mathbf{x}} + \alpha\mathbf{x}) \approx \Omega(\widetilde{\mathbf{x}}) + \alpha\langle\nabla\Omega(\widetilde{\mathbf{x}}), \mathbf{x}\rangle.$$

En prenant pour $\mathbf{x}$ le gradient normalisé $\frac{\nabla\Omega(\widetilde{\mathbf{x}})}{\|\nabla\Omega(\widetilde{\mathbf{x}})\|}$ il vient

$$\Omega(\widetilde{\mathbf{x}} + \alpha\frac{\nabla\Omega(\widetilde{\mathbf{x}})}{\|\nabla\Omega(\widetilde{\mathbf{x}})\|}) \approx \Omega(\widetilde{\mathbf{x}}) + \alpha\|\nabla\Omega(\widetilde{\mathbf{x}})\|).$$

En définissant une fonction F par

$$F(\alpha) = \frac{\Omega\left(\widetilde{\mathbf{x}} + \alpha\frac{\nabla\Omega(\widetilde{\mathbf{x}})}{\|\nabla\Omega(\widetilde{\mathbf{x}})\|}\right) - \Omega(\widetilde{\mathbf{x}})}{\alpha\|\nabla\Omega(\widetilde{\mathbf{x}})\|},$$

alors F doit vérifier $\lim_{\alpha\to 0} F(\alpha) = 1$.

La figure (5.3) représente la valeur absolue $|F(\alpha) - 1|$ pour différentes valeurs de α. Nous observons que la différence converge bien vers 0 avec une précision de l'ordre de 10^{-4} / 10^{-5}. Ceci confirme la validité de l'implémentation du gradient. Les irrégularités de $|F(\alpha) - 1|$ observées pour des valeurs de α comprises entre $(\frac{1}{2})^{28}$ et $(\frac{1}{2})^{32}$ sont dues au fait que, pour ces valeurs, la précision machine est atteinte.

Test du produit scalaire

Pour vérifier que le code adjoint est bien l'adjoint du linéaire tangent, nous choisissons deux vecteurs $d\mathbf{x}$ et $d\mathbf{y}^*$ et nous calculons

$$d\mathbf{y} = \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}\right) \cdot d\mathbf{x}$$
$$d\mathbf{x}^* = \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}\right)^* \cdot d\mathbf{y}^*.$$

Or

$$\begin{aligned}\langle d\mathbf{y}^*, d\mathbf{y}\rangle &= \langle d\mathbf{y}^*, \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}\right) \cdot d\mathbf{x}\rangle \\ &= \langle \left(\frac{\partial \mathcal{M}}{\partial \mathbf{x}}\right)^* \cdot d\mathbf{y}^*, d\mathbf{x}\rangle \\ &= \langle d\mathbf{x}^*, d\mathbf{x}\rangle.\end{aligned}$$

Le test consiste alors à vérifier (à la précision machine) que $\langle d\mathbf{x}^*, d\mathbf{x}\rangle = \langle d\mathbf{y}^*, d\mathbf{y}\rangle$.

Comme autre étape de vérification du modèle tangent, nous pouvons simplement appliquer cet opérateur aux vecteurs de la base canonique de $\mathbb{R}^n$, puis vérifier que nous obtenons bien la matrice transposée du modèle direct appliquée aux mêmes vecteurs. De même les codes des inverses des matrices de corrélations sont testés en vérifiant (exemple pris pour $\mathbf{B}$) : $\frac{\mathbf{x}-\mathbf{B}(\mathbf{B}^{-1}(\mathbf{x}))}{\|\mathbf{x}\|} \approx \frac{\mathbf{x}-\mathbf{B}^{-1}(\mathbf{B}(\mathbf{x}))}{\|\mathbf{x}\|} \leq 10^{-12}$.

5.2 Minimisation du 4DVar pénalisé

Dans ce nouveau contexte nous mettons en oeuvre les différents algorithmes présentés. Nous allons montrer l'efficacité des algorithmes de type dual proposés par rapport à ceux déjà existants, et notamment par rapport aux algorithmes dans les espaces de Hilbert. Le 4DVar à pénaliser est toujours de la forme

$$\Omega_p(\mathbf{x}) = \frac{1}{2}\|x - x_0^b\|^2_{\mathcal{B}^{-1}} + \frac{1}{2}\|\mathbf{y} - \hat{\mathcal{H}}x\|^2_{\mathbf{R}^{-1}} + \frac{\lambda}{p}\|\mathbf{\Phi x}\|_p^p \tag{5.5}$$

Nous pouvons distinguer deux classes d'algorithmes permettant de minimiser cette fonctionnelle : la première consiste en ceux linéarisant les opérateurs non linéaires dans une boucle externe avant de minimiser la fonctionnelle résultante dans une boucle interne, la seconde regroupe ceux minimisant directement (5.5) (mais qui utiliseront une boucle pour la recherche de pas). À des fins d'équité et dans un soucis de réalisme (le nombre d'itérations dans une configuration d'assimilation de données étant limité), nous octroyons aux algorithmes de la première classe deux boucles externes comprenant quinze boucles internes chacune, et pour les algorithmes de la seconde classe un nombre total de trente itérations.

5.2.1 Choix de la base Φ pour la pénalisation

Plusieurs stratégies existent pour le choix d'une base pour le terme de régularisation. Elles dépendent de la quantité d'informations que possède l'utilisateur regardant la structure de la variable pénalisée. À la vue du véritable état initial de la première fenêtre (5.1a), il paraît judicieux de faire porter la pénalisation dans une base de la dérivée (numérique) du signal. Cependant, à cause du phénomène de lissage induit par le modèle, cette même base ne serait pas adéquate pour la seconde fenêtre (5.1b). Nous n'avons pas introduit d'erreur modèle dans ce système d'assimilation, mais il serait tout à fait possible que la structure quasi-parcimonieuse du véritable état initial soit conservée au cours du temps. Réutiliser une base de la dérivée serait une manière d'injecter cette information dans la fonctionnelle du 4DVar.

Les bases d'ondelettes permettent de représenter sous une forme parcimonieuse une grande partie des signaux physiques rencontrés usuellement. En particulier, une base d'ondelettes (orthogonales) de Daubechies d'ordre 2 est pertinente si un utilisateur s'attend à obtenir des signaux linéaires par parties, ou présentant certains fronts tels que sur la Figure (5.1a). En effet, il s'agit d'une base possédant deux moments dissipants, c'est à dire qu'elle représente de manière parcimonieuse les polynômes d'ordre inférieur ou égaux à 1. **C'est ce choix que nous ferons pour la suite des expériences.** Nous verrons qu'en particulier ce choix permet de retrouver un état analysé de bonne qualité pour la seconde fenêtre, même si l'état attendu a été lissé. Une panoplie d'autres choix sont possibles (cf. ([32]) qui doivent être considérés selon le cadre d'application.

La régularisation porte ici uniquement sur la hauteur du fluide z. La matrice $\mathbf{\Phi}$ consiste donc en deux blocs de $\mathbf{0}$ concaténés horizontalement (pour ne pas pénaliser u et v), suivi d'un bloc opérant la projection orthogonale dans la base de Daubechies 2 de z.

5.2.2 Choix des paramètres λ et p

Comme la modélisation du système ne suppose pas d'erreurs gaussiennes généralisées, nous sommes dans le cadre d'une régularisation en norme L_p motivée par la structure du problème. Nous n'avons donc pas de valeur indiquée pour p ni pour λ. Leur détermination peut reposer, comme pour l'expérience d'advection, sur le principe de Morozov ou de la L-curve. Toutefois, les résultats du chapitre (3) portent à préférer une valeur de p proche de 1 pour le type de signal quasi-creux rencontré.

En accord avec la limite inférieure de p présentée à la section (4.2.2), nous commençons par estimer la valeur seuil p_{min} qui rendrait numériquement instable les algorithmes duaux. Nous pouvons estimer que la hauteur du fluide ne dépassera pas, par rapport à l'altitude de référence, la valeur $M = 300\text{m}$. Les valeurs de p possibles sont donc supérieures à ≈ 1.01.

En outre, nous sommes également intéressé par la dépendance de la RMSE obtenue au terme du nombre d'itérations allouées en fonction de ces deux paramètres. Nous dressons donc une carte de chaleur de la RMSE, ce qui permettra de voir sa sensibilité par rapport à λ et p et de choisir un couple de valeur pour effectuer l'ensemble des expériences. La carte de chaleur pour l'algorithme de gradient conjugué non linéaire dans l'espace dual est représenté sur la Figure (5.4). Nous voyons que la régularisation est robuste vis-à-vis de p pour des valeurs de λ faible et, inversement, pour des valeurs de p proche de 1 la RMSE est stable vis-vis des variations de λ. Une dégradation des résultats se fait sentir lorsque p et λ augmentent simultanément dans la partie supérieur droite du tableau.

Au regard de ces résultats, **nous choisissons arbitrairement parmi les couples (λ,p) acceptables les valeurs $p = 1.1$ et $\lambda = 1$ pour l'ensemble des expériences.**

5.2.3 Vitesse de convergence des algorithmes

Plus faible erreur obtenue au terme des itérations

À l'instar des comparaisons effectuées précédemment, nous commençons par dresser une table de la RMSE et de la MAE à $t = 0$ et à $t = T$ obtenues au terme du nombre d'itérations allouées pour chaque algorithme. Ces erreurs portent donc sur la différence entre l'état analysé et l'état vrai pour le début de chaque fenêtre d'assimilation. De nouveau, pour prendre en compte l'aléatoire survenant lors des observations, 20 expériences sont lancées pour chaque algorithme et affichées sur la Figure (5.5a) pour la première fenêtre et sur la figure (5.5b) pour la seconde fenêtre.

Un groupe de points noirs correspondant aux résultats obtenus par l'algorithme du gradient conjugué d'Estatico *et al.* (qui est une généralisation de l'algorithme du gradient conjugué dans les espaces de Banach, cf. Section 2.2.3) se distingue au premier regard par ses résultats décevants sur les deux fenêtres. Cet algorithme requiert le calcul d'un pas appartenant à un certain intervalle, calculé en pratique par un algorithme conçu pour l'optimisation avec contraintes sans dérivée ([91]) basée sur une méthode itérative de régions de confiance. Il s'avère qu'en pratique le pas calculé est très petit. Les itérés changent ainsi très peu et ne se rapprochent que très lentement d'un minimum. Dans son état actuel, cet algorithme ne semble donc pas adapté à la résolution de problèmes d'assimilation. Nous montrons pourtant en Annexe (A) qu'une technique de backtracking permettrait des pas plus grands, bien qu'elle n'aurait alors pas de garantie théorique de convergence.

0. Les abréviations utilisées sont listées ici : **(R)PCG** : (Restricted) Preconditioned Conjugate Gradient / Gradient conjugué préconditionné ; **NLCG** : Non-Linear Conjugate Gradient / Gradient conjugué non linéaire ; **NLCGDS** : Non-Linear Conjugate Gradient Dual Space / Gradient conjugué non linéaire dans l'espace dual ; **CGB** : Conjugate Gradient Banach / Gradient conjgué d'Estatico *et al.* ; **GDDS** : Gradient Descent Dual Space / Descente de gradient dans l'espace dual.

Une nette amélioration des erreurs pour la première fenêtre se produit successivement entre l'algorithme de descente de gradient classique (en rouge) et l'algorithme de descente de gradient dans l'espace dual (en violet), puis entre ce dernier algorithme et celui de gradient conjugué non linéaire dans l'espace dual (en bleu foncé), soulignant l'intérêt des ces algorithmes dans ce contexte.

L'algorithme du gradient conjugué préconditionné RPCG ([59]) (en rose) a un statut particulier : cet algorithme n'est initialement pas conçu pour prendre en compte une régularisation en norme L_p. S'il est alors plus difficile d'interpréter la RMSE obtenue par cet algorithme par rapport aux autres (la fonctionnelle minimisée n'étant alors plus la même), il sert quand même de référence en terme de qualité des solutions obtenues et va permettre de mettre en valeur les bénéfices d'avoir utilisé une telle régularisation. Il bat ici les algorithmes de premier ordre mais se trouve derrière l'algorithme de gradient conjugué non linéaire dans le dual.

Les résultats pour la seconde fenêtre d'assimilation sont plus homogènes. Mis à part les points noirs, l'algorithme de la descente de gradient classique se trouve derrière un agrégat de points similaires obtenus par les trois autres algorithmes. En fait, la solution attendue pour la seconde fenêtre est lisse et ne possède plus de parcimonie particulière dans la base d'ondelettes choisie. Néanmoins, la régularisation dans cette base ne dégrade pas pour autant le résultat. Ainsi, l'algorithme de Gauss-Newton utilisant RPCG dans la boucle interne réalise une aussi bonne performance que les algorithmes duaux.

Comparaison graphique

Une visualisation des états analysés pour la première fenêtre d'assimilation (à $t = 0$s) permet de mieux se rendre compte des performances des divers algorithmes. La Figure (5.6) compare ces états pour le gradient conjugué non linéaire dans le dual (Figure (5.6a)) et la descente de gradient dans le dual (Figure (5.6b)). Les résultats quantitatifs de la Figure (5.5) se retranscrivent ici par la meilleure analyse de NLCGDS vis-à-vis du vrai état initial (5.1a).

Nous montrons sur la Figure (5.7a) comment se comporte la courbe de décroissance de la fonctionnelle lorsqu'elle est minimisée par les deux algorithmes susmentionnés. Dès la cinquième itération, NLCGDS obtient une plus faible erreur que GDD. Cette décroissance se répercute sur l'évolution, au cours des itérations, de la RMSE entre l'itéré courant et $\mathbf{x}_{true}$ comme le montre la Figure (5.7b).

Les figures (5.8a) et (5.8b) permettent de mesurer l'intérêt de la régularisation en norme L_p en comparant le profil de l'état analysé, toujours à $t = 0$s, obtenu pour NLCGDS (qui minimise le 4DVar pénalisé) et RPCG (qui minimise le 4DVar sans pénalisation). À la fin des itérations, NLCGDS a permis de retrouver une structure en plateaux plus conforme à la vraie solution initiale là ou RPCG a convergé vers une structure plus lisse.

Il est également intéressant de comparer ces deux derniers algorithmes lorsqu'ils minimisent la même fonctionnelle. Nous prenons donc temporairement $\lambda = 0$. La figure (5.9) montre à nouveau l'état analysé produit par ces algorithmes. NLCGDS a clairement mené à la solution la plus satisfaisante. Nous remarquons que, même sans terme de pénalisation explicite, la solution calculée par NLCGDS présente toujours une structure plus proche de l'état vrai. Même en accordant plus d'itérations à RPCG, celui-ci continuerait à renvoyer une solution lisse en forme de bosse. Nous conjecturons que l'opérateur de dualité permet une forme de régularisation implicite sur les itérés. Cette conjecture est illustrée plus en avant dans la seconde section de l'annexe B.

Remarquons enfin que pour ce système, puisqu'aucune erreur modèle n'a été considérée et que l'état perd sa parcimonie initiale au cours des itérations, obtenir une meilleure RMSE à $t = 0$s n'est pas équivalent à obtenir une meilleure RMSE sur toute la fenêtre. Ce phénomène est illustré sur la Figure (5.10) qui affiche la RMSE au cours du temps (c'est à dire la RMSE entre l'état analysé propagé par le modèle numérique jusqu'à t et l'état vrai au temps t) produite par NLCGDS (avec régularisation) et RPCG. Nous y voyons un cas favorable pour NLCGDS (Figure (5.10a)) et défavorable (Figure (5.10b)). Pour obtenir la deuxième figure, nous avons pris $T = 43200$s au lieu de $T = 21600$s. En effet, plus la période d'assimilation est grande, plus l'erreur modèle va être importante et plus la parcimonie sera perdue entre deux assimilations. L'impact de la régularisation sur la condition initiale sera également atténué. Par suite, un système d'assimilation qui ne prendrait pas en compte l'erreur modèle et où des états parcimonieux sont attendus nécessite des cycles d'assimilation d'autant plus courts.

5.2.4 Comportement de NLCGDS

Nombre d'itérations requis pour la recherche de pas

Si NLCGDS ne nécessite pas deux boucles imbriquées pour minimiser le 4DVar, il nécessite néanmoins une étape de recherche de pas qui constitue le coût principal en nombre d'opérations de l'algorithme. Il fait appel à chaque tour de boucle de la recherche de pas à l'évaluation de Ω en un point pour la condition d'Armijo (modifiée) seule et en plus à l'évaluation de $\nabla\Omega$ pour les conditions de Wolfe. Il est donc intéressant de regarder si d'une part les deux conditions de Wolfe sont nécessaires pour converger rapidement et, d'autre part, de regarder le nombre de boucles requises.

Sur la totalité des expériences menées, la seconde condition de Wolfe n'a jamais joué (autrement dit elle était toujours vérifiée) lors des 30 itérations avec $c_1 = 10^{-3}$ et $c_2 = 0.9$. D'après les résultats précédents il semble donc, au moins sur cet exemple, que même sans cette condition NLCGDS permet une convergence rapide. Bien qu'il manque encore une preuve de convergence de l'algorithme avec condition d'Armijo modifiée seule, il semble souhaitable d'utiliser cette condition pour la recherche de pas en assimilation de données.

Par rapport au nombre d'itérations effectuées dans la boucle de la recherche de pas, les résultats sont donnés pour le pas initial $\mu_0 = 1$ dans la Table (5.1). Le nombre total requis à la fin de l'expérience, de l'ordre de plusieurs centaines, peut être largement réduit en choisissant un pas initial plus ingénieusement. Par exemple une stratégie possible serait de prendre $\mu_0 = 1$ pour la première itération, puis, en supposant que l itérations lors de la recherche de pas aient été requises pour cette itération, prendre $\mu_0 = (\frac{1}{2})^l$ pour la deuxième recherche linéaire. Cette stratégie simple permet sur l'exemple de la Table (5.1) de ramener le nombre total d'itérations à 330.

TABLE 5.1 – Nombre de tours de boucle requis pour la recherche de pas au cours des itérations de GDD et NLCGDS (première fenêtre d'assimilation).

Itération	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
GDD	11	14	13	14	14	14	13	14	14	13	14	14	14	13	14	14	14	13	14	14
NLCGDS	11	15	14	14	14	14	14	14	14	13	14	14	14	14	14	13	14	14	14	14

Itération	21	22	23	24	25	26	27	28	29	30	Nombre total de tours de boucle
GDD	14	13	14	14	14	13	14	14	14	13	409
NLCGDS	13	14	14	14	14	13	14	14	14	14	414

Utilisation du garde-fou

Nous nous demandons à présent s'il a été nécessaire de faire appel au garde-fou introduit à la section (4.2.2). Nous comptons de nouveau le nombre de fois que le garde-fou s'est déclenché, i.e. $H_k(\mathbf{x}_k^*) = \nabla f_k$, au cours des itérations. Comme attendu, pour un nombre restreint d'itérations, le garde-fou n'a pas le temps de se déclencher : il ne s'est activé aucune fois sur l'ensemble des expériences menées que ce soit sur la première ou la seconde fenêtre d'assimilation. Cette observation confirme que cette étape, nécessaire à la preuve de convergence de l'algorithme, n'est pas indispensable en pratique.

Nombre de restarts effectués

L'étape de backtracking sur β_k, pour s'assurer que p_k est bien une direction de descente pour f_k et $(f \circ J_q)(\mathbf{x}_k^*)$, demande d'évaluer le gradient de ces deux quantités. Il faut de toutes façons calculer ∇f_k, mais nous ne disposons de $\nabla(f \circ J_q)(\mathbf{x}_k^*)$ que si la seconde condition de Wolfe a été utilisée pour la recherche du pas. Une fois ces deux vecteurs de taille n calculés, chaque tour de boucle de backtracking demande alors $2(n + (n-1)) = 4n - 2$ opérations. Nous fixons ici le nombre maximal de tour de boucle avant un restart forcé à $n^{\beta}_{itermax} = 5$.

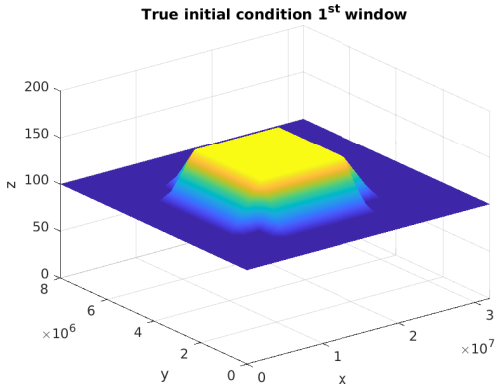
De nouveau nous observons des résultats encourageants : aucun restart n'a été effectué. Devant le faible coût de cette stratégie il est cependant prudent de la conserver, au cas où

une direction serait malencontreusement mauvaise et dégraderait la qualité des directions successives.

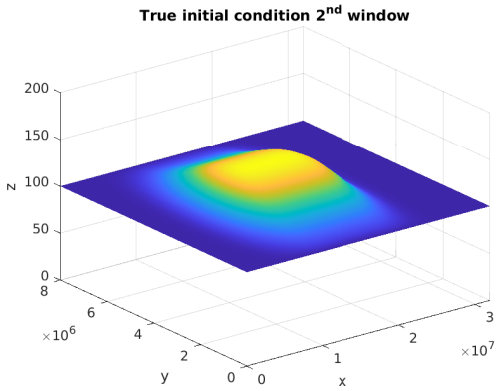
5.3 Conclusion

Cette configuration plus réaliste d'un système d'assimilation basées sur les équations de Barré de Saint-Venant a permis le passage à l'échelle des observations faites au chapitre (3). D'abord, la régularisation en norme L_p s'est de nouveau montrée efficace pour retrouver un signal quasi-creux (première fenêtre d'assimilation considérée). De plus, en faisant porter la pénalisation sur le signal projeté dans une base d'ondelettes de Daubechies, la solution obtenue n'a pas souffert de la régularisation alors que la structure lisse du signal attendu (deuxième fenêtre d'assimilation) ne justifiait pas *a priori* une pénalisation en norme L_p. Si la parcimonie du signal est conservée au cours du temps par le modèle numérique, la régularisation en norme L_p permet de compenser l'absence d'un terme d'erreur modèle dans le 4DVar, mais la durée des fenêtres d'assimilation doit être suffisamment courte pour que cette erreur ne devienne pas prépondérante à cause d'un lissage du signal.

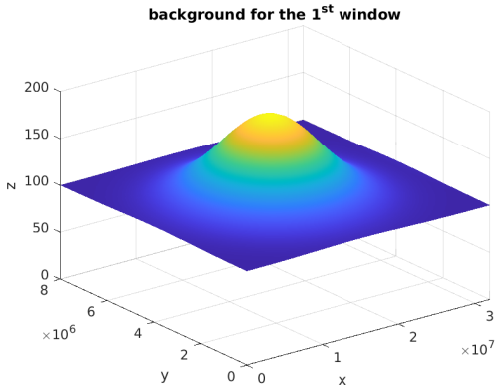
Ensuite, ce système a permis de montrer l'efficacité des algorithmes duaux proposés, en particulier le gradient conjugué non linéaire dans le dual, capable de minimiser le 4DVar efficacement avec et sans pénalisation. Nous avons pu observer que ce dernier algorithme ne requérait pas, dans ce contexte, les détails techniques du chapitre (4) (garde-fou, restart), mais sa complexité en temps de calcul dépend d'un choix judicieux d'un pas initial pour la recherche du pas.



(a) Véritable condition initiale pour la première fenêtre de l'assimilation.



(b) Véritable condition initiale pour la seconde fenêtre de l'assimilation.



(c) Ébauche gaussienne pour la première fenêtre de l'assimilation.

FIGURE 5.1 – Conditions initiales pour les deux fenêtres de l'assimilation et ébauche pour la première fenêtre de l'assimilation (x,y et z en mètres).

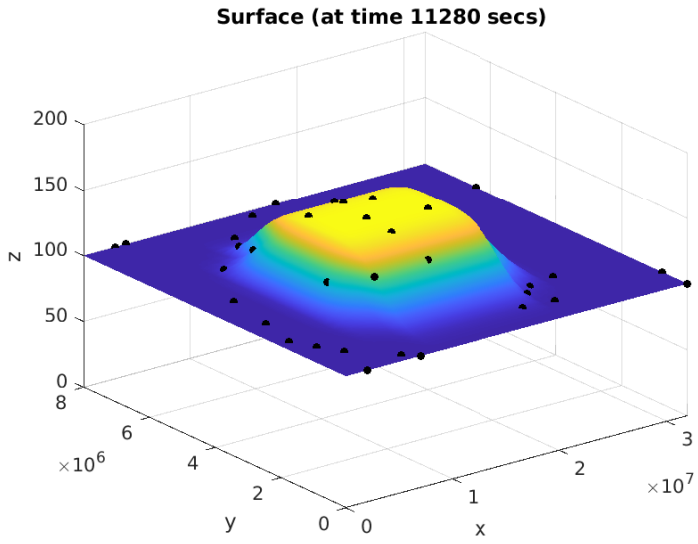


FIGURE 5.2 – Exemple d'un scénario d'observation à un instant fixé : les points où l'altitude est relevée sont indiqués en noir, puis un bruit gaussien y est ajouté.

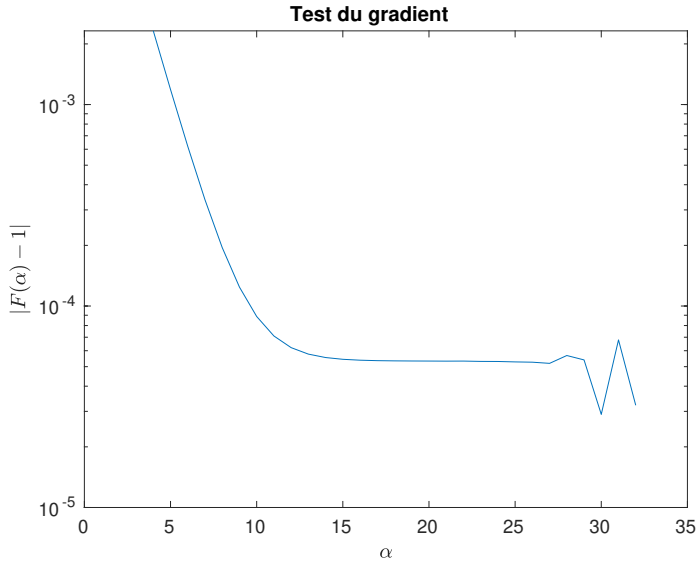


FIGURE 5.3 – Test du gradient. En ordonnée : $|F(\alpha) - 1|$ en échelle logarithmique. En abscisse : $\alpha = (\frac{1}{2})^i$ avec $4 \leq i \leq 32$.

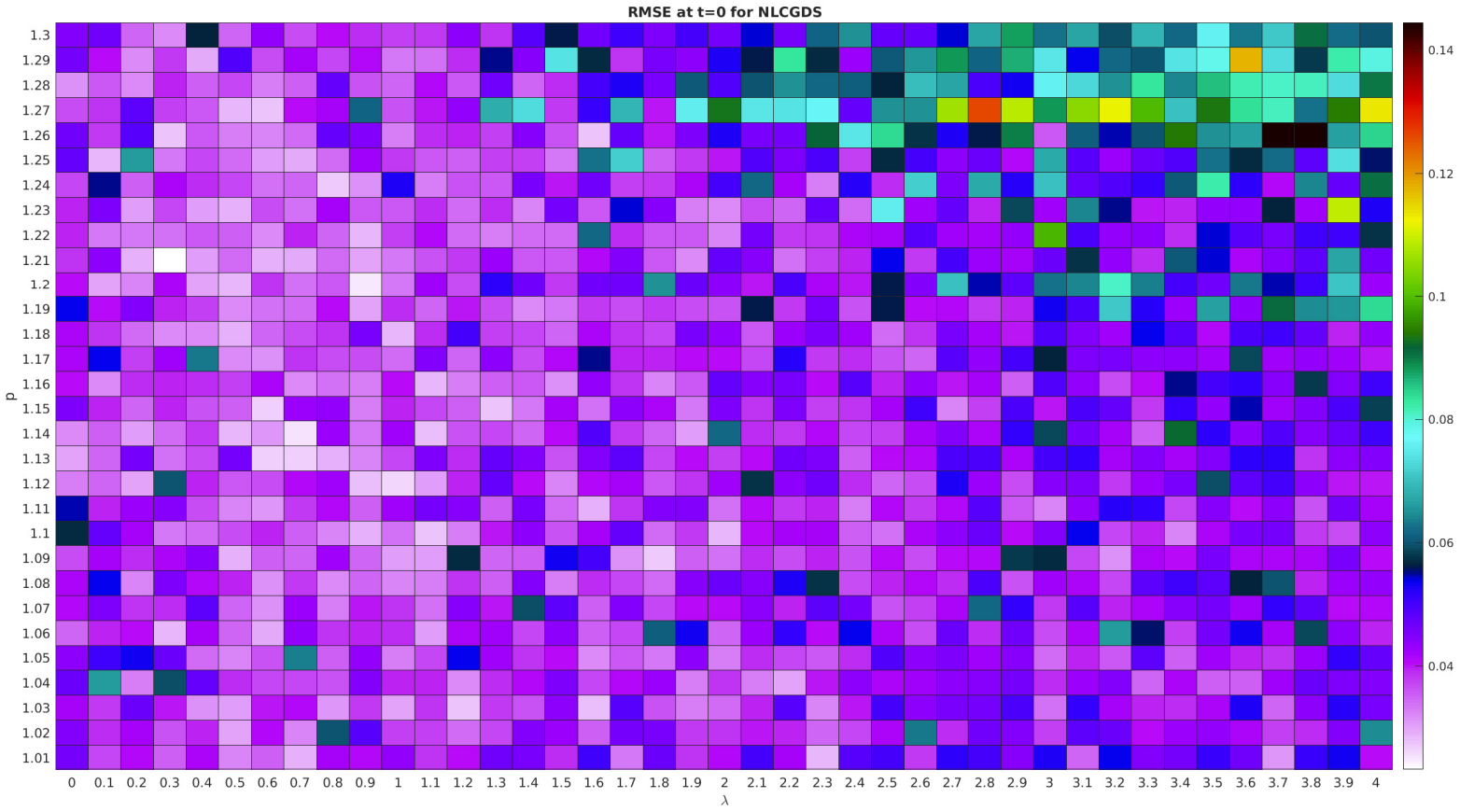
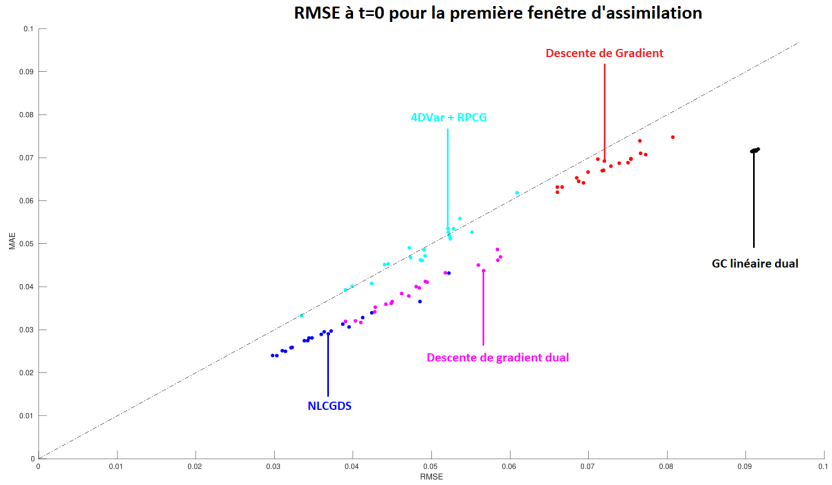
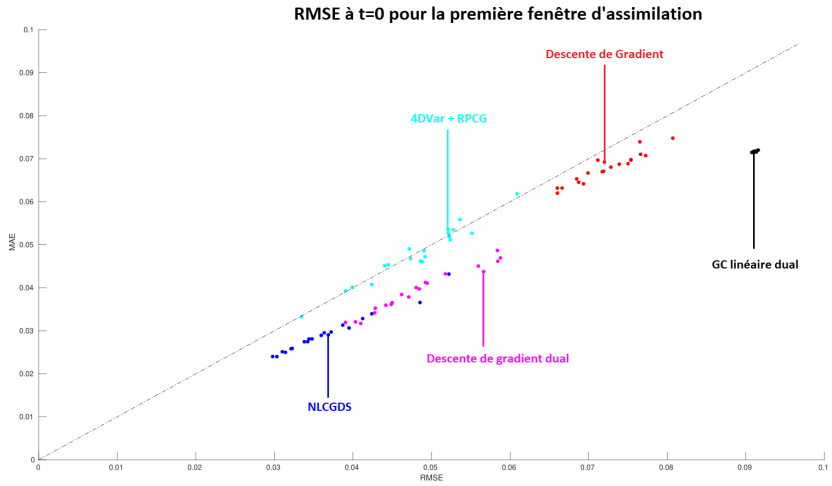


FIGURE 5.4 – Carte de chaleur de la RMSE entre l'état analysé et le vrai état initial au temps $t = 0$ en fonction de λ et p.

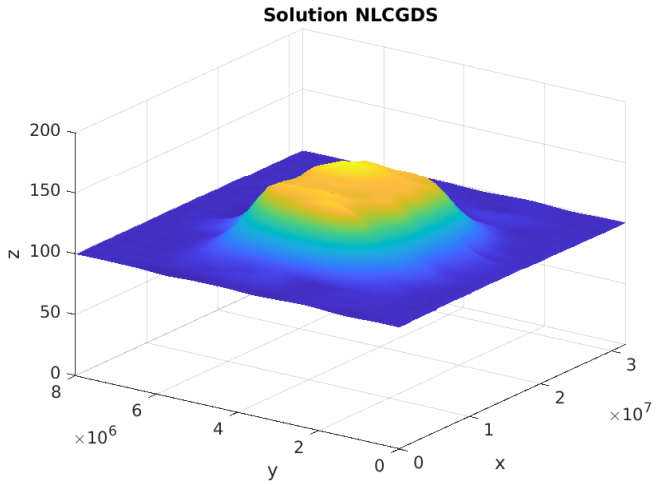


(a)

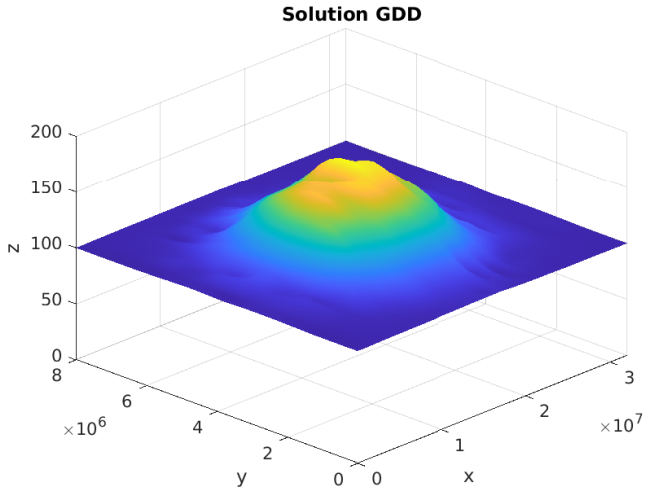


(b)

FIGURE 5.5 – RMSE/MAE pour 20 expériences entre l'état analysé des différents algorithmes et l'état vrai de la première fenêtre d'assimilation (a) et de la seconde fenêtre (b). En noir : l'algorithme d'Estatico, en rouge : la descente de gradient classique, en bleu clair : la descente de gradient dans le dual, en rose : l'algorithme RPCG, en bleu foncé : le gradient conjugué non linéaire dans le dual.

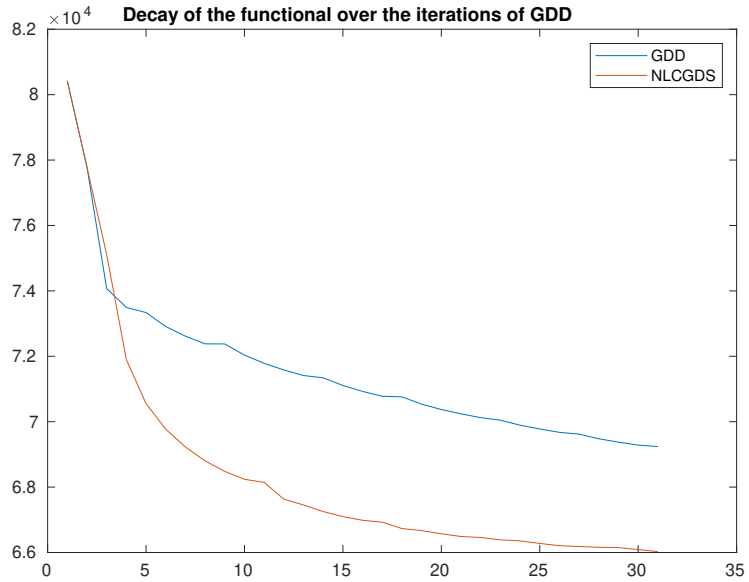


(a) État analysé pour le gradient conjugué non linéaire dans le dual.

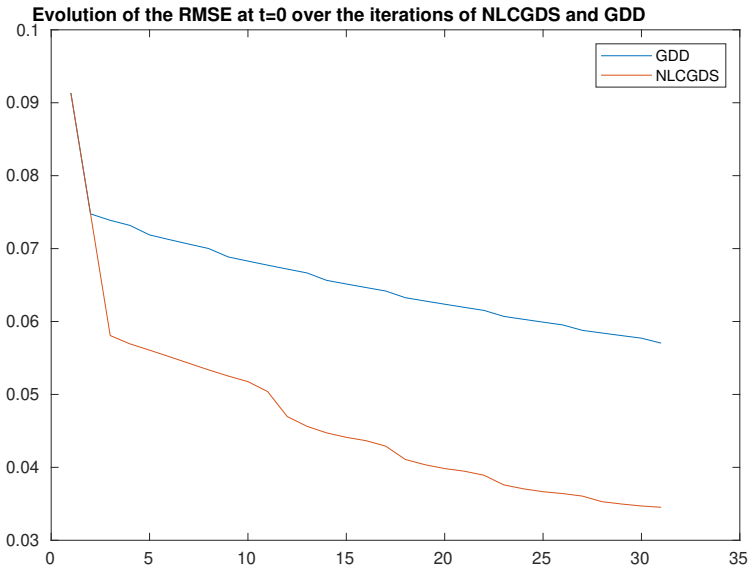


(b) État analysé pour la descente de gradient dans le dual.

FIGURE 5.6 – État analysé pour la descente de gradient dans le dual et le gradient conjugué non linéaire dans le dual (pour la première fenêtre d'assimilation).

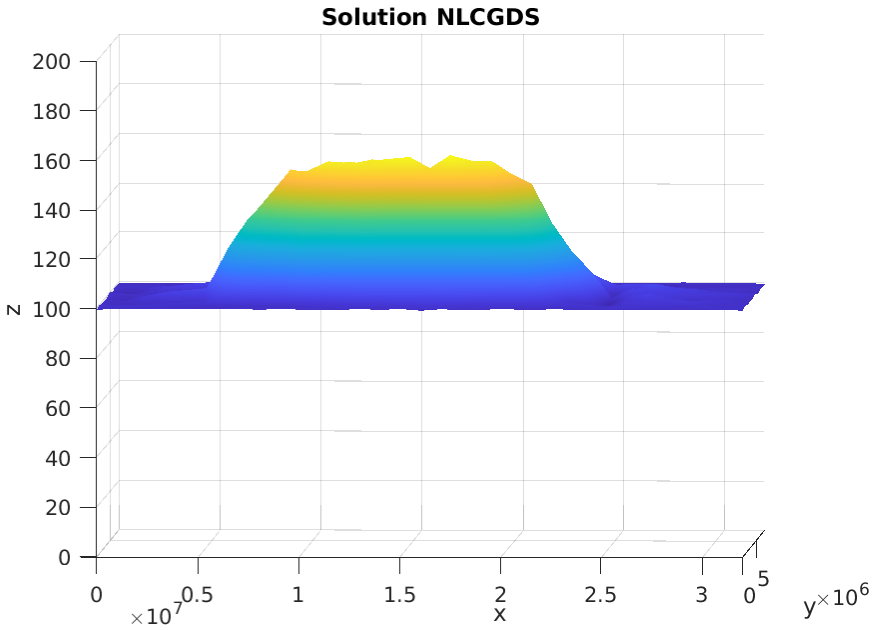


(a) Décroissance de la fonctionnelle du 4DVar pénalisée au cours des itérations de la descente de gradient dans le dual (GDD en bleu) et du gradient conjugué non linéaire dans le dual (NLCGDS en orange).

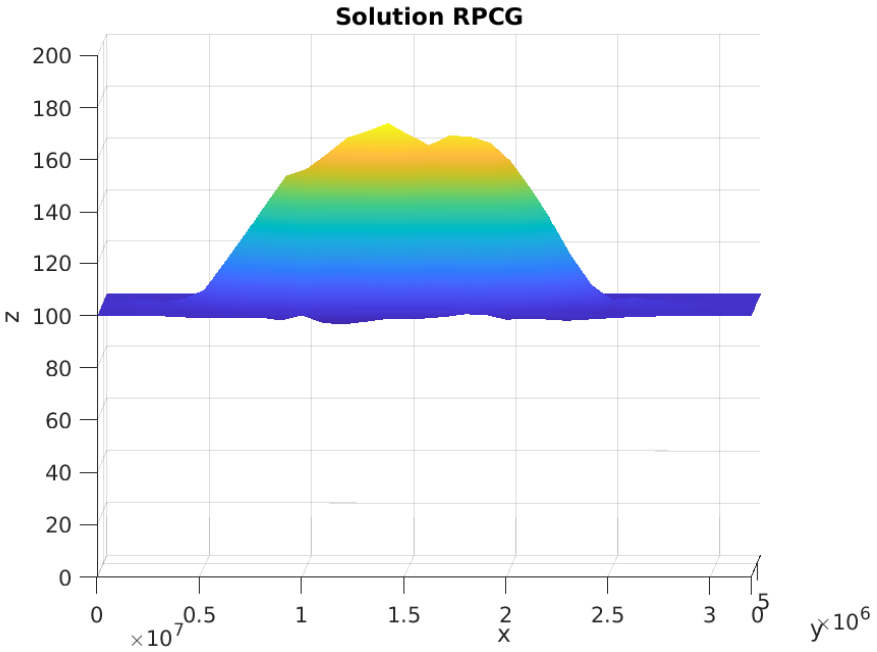


(b) RMSE at $t = 0$ (en ordonnée) entre les itérations de GDD (en bleu) et de NLCGDS (en orange) au cours des itérations (en abscisse).

FIGURE 5.7 – Comparaison entre GDD et NLCGDS.

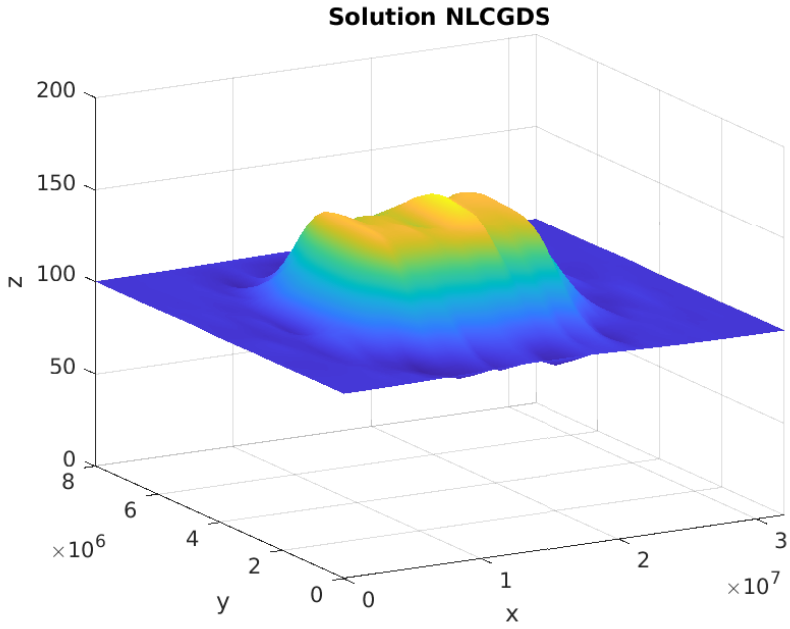


(a) Profil de l'état analysé pour le gradient conjugué non linéaire dans le dual (première fenêtre d'assimilation).

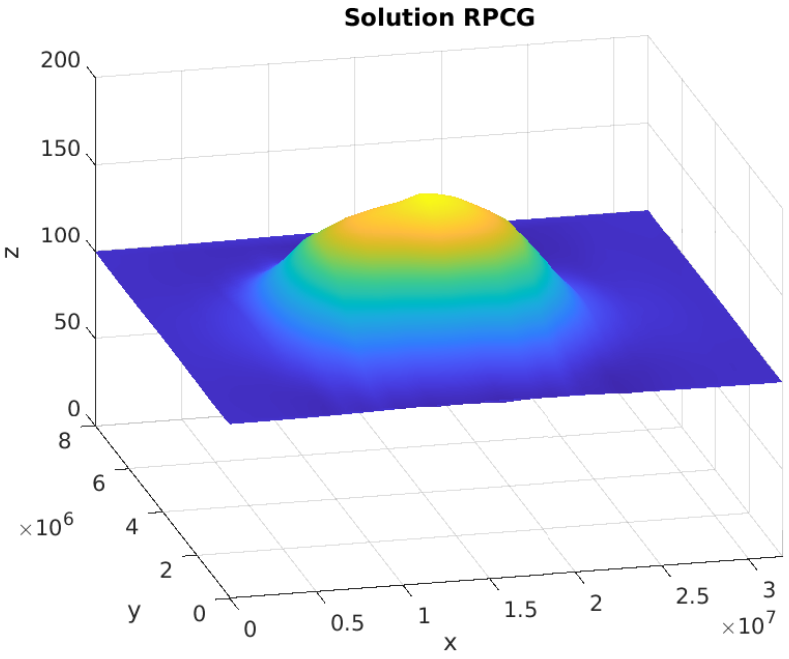


(b) Profil de l'état analysé pour RPCG (première fenêtre d'assimilation).

FIGURE 5.8 – Profils des états analysés par NLCGDS et par RPCG pour la première fenêtre d'assimilation (x,y et z en mètres).

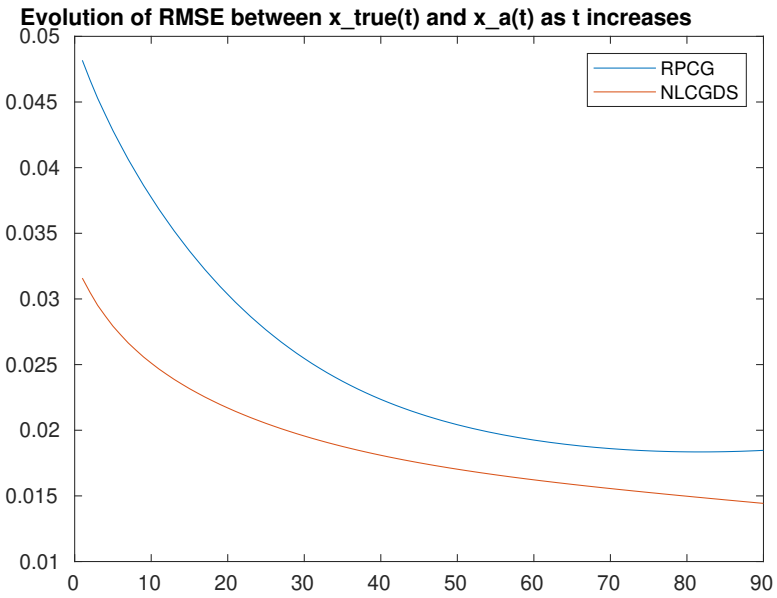


(a) État analysé pour le gradient conjugué non linéaire dans le dual (première fenêtre d'assimilation).

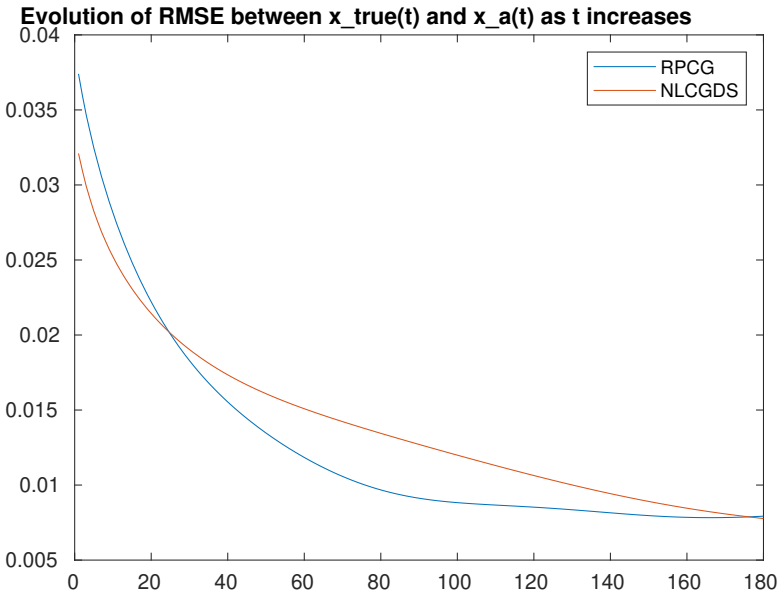


(b) État analysé pour RPCG (première fenêtre d'assimilation).

FIGURE 5.9 – Comparaison entre NLCG avec $\lambda = 0$ et RPCG (x,y et z en mètres).



(a) RMSE entre l'état analysé propagé par le modèle et le véritable état au cours du temps pour la première fenêtre d'assimilation ($T = 21600$s).



(b) Une RMSE plus petite à $t = 0$ pour l'un des algorithmes n'implique pas une RMSE plus petite pour la première fenêtre d'assimilation ($T = 43200$s).

FIGURE 5.10 – Évolution de la RMSE au cours du temps entre l'état analysé propagé par le modèle pour RPCG (en bleu) et NLCGDS (en orange) et le véritable état initial (abscisse : temps mesuré en Δt, ordonnée : RMSE).

Conclusion et perspectives

Bilan des contributions

Nous nous sommes d'abord posé la question de la possibilité de considérer des fonctions de répartition différentes de la loi normale pour modéliser des phénomènes physiques. En effet, de nombreux exemples ont montré que la loi normale n'était pas la distribution la plus adéquate pour décrire le comportement de toutes les variables rencontrées dans la réalité. Son extension la plus immédiate, la loi gaussienne généralisée, amène, dans un contexte bayésien, à considérer la minimisation d'un terme en norme L_p. Différentes approches font porter la norme L_p soit directement sur le terme d'écart à l'ébauche, soit sur un terme additionnel de pénalisation de la variable d'état (potentiellement dans une autre base que la base canonique).

La norme L_p en tant que régularisation s'est avérée posséder des avantages intrinsèques, indifféremment de la modélisation statistique. Elle permet en effet de sélectionner numériquement une solution avec certaines propriétés : elle s'est avérée pertinente pour retrouver des signaux quasi-parcimonieux présentant des fronts brusques mais continus que l'on trouve effectivement dans des cas pratiques d'assimilation de données (fronts météorologiques, épaisseur de la glace, concentration de la sargasse dans la mer etc.). Nous avons également vu que cette norme permettait d'atténuer les plateaux et les oscillations non désirables introduits numériquement sur la solution, respectivement par les régularisations en norme L_1 et L_2.

Les intérêts quant à l'utilisation de cette norme ayant été attestés, il a été nécessaire de trouver un moyen pour minimiser efficacement le 4DVar ainsi pénalisé. Guidés par le cadre mathématique des espaces non-euclidiens découlant de l'utilisation de la norme L_p, nous nous sommes tournés vers des algorithmes reposant sur le transport des itérations ou des directions entre l'espace primal de la variable d'état et son dual topologique, issus de la littérature sur la minimisation dans les espaces de Banach. Nous avons modifié l'algorithme de descente de gradient avec transport des itérations dans le dual en ajoutant une recherche de pas basée sur les conditions d'Armijo et de Wolfe s'effectuant dans l'espace dual. Ces algorithmes ont servi de fondement pour proposer deux algorithmes de gradient conjugué dans l'espace dual (à nouveau l'un avec transport des itérés, l'autre

avec transport de la direction). Leur efficacité pour prendre en compte le nouveau terme de régularisation a été démontrée expérimentalement, et l'algorithme de gradient conjugué avec transport des itérés s'est démarqué en terme de robustesse et de rapidité de convergence vis-à-vis des choix de p et du paramètre de régularisation λ. En revanche, les algorithmes avec transport de la direction sont plus sensibles pour des valeurs de p proches de 1 et ce d'autant plus que la norme des gradients est élevée, ce qui n'en font pas de bons candidats pour l'assimilation de données.

Pour démontrer la convergence de ces nouveaux algorithmes, nous avons repris la condition de convergence de Zoutendijk que nous avons étendue aux algorithmes faisant usage de l'espace dual. Nous nous sommes intéressés au choix particulier de $p_k = -\nabla f_k$, sachant que les nouvelles conditions de Zoutendijk sont valables pour un choix de p_k quelconque. Les nouvelles conditions d'angles à vérifier pour assurer la convergence font intervenir le crochet de dualité entre $J_q(p_k)$ ou $J'_q(x^*_k)p_k$ et $-\nabla f_k$, normalisés par les normes L_p et L_q correspondantes.

Enfin, les tests effectués sur le modèle des équations de Navier-Stokes en eau peu profonde ont permis de confirmer le caractère prometteur de la régularisation et des algorithmes proposés pour un usage réel. De fait, la détermination de la structure correcte du véritable état initial pour le nombre d'itérations alloué n'a eu lieu que grâce à l'utilisation du gradient conjugué non linéaire dans le dual. Ces expériences ont permis de retrouver les résultats que l'expérience simple d'advection unidimensionnelle avait mis en avant (en particulier la capacité de la norme L_p à améliorer la reconstruction d'un signal quasi-parcimonieux malgré la diffusion, soit d'origine physique soit numérique), mais sur un modèle plus complexe et de plus grande dimension. Elles nous confortent donc dans l'idée que les résultats obtenus sont généraux et donc que la stratégie de pénalisation présentée est applicable à d'autres problèmes d'assimilation de données.

Perspectives

Les perspectives pour approfondir le travail effectué sont multiples. Comme mentionné à l'instant, il manque avant tout des expériences sur des données réelles, qui auraient peut-être dévoilées des difficultés nouvelles. Nous aurions aimer appliquer la régularisation et les algorithmes au cas de la glace de mer, dont l'évolution dépend d'un système couplé océan/glace et qui nécessite à l'heure actuelle le développement numérique d'un modèle adjoint. Il semble également intéressant d'utiliser cette pénalisation dans le cas de la détermination non plus d'un état du système, mais de paramètres intervenant dans les équations physiques. En effet, les motivations initiales restent toujours valides si l'on s'attend à ce que ces paramètres possèdent une structure spatiale ou temporelle parcimonieuse ou dont la distribution suivrait une loi gaussienne généralisée.

L'assimilation de données basée sur les équations de Barré de Saint-Venant a montré que, si la pénalisation permettait de mieux reconstruire l'état initial, le caractère diffu-

sif des équations faisait perdre à l'état analysé sa structure parcimonieuse au cours du temps. Nous avons suggéré d'utiliser des fenêtres d'assimilation courtes afin de pouvoir maintenir cette structure mais une autre possibilité serait d'ajouter un terme de pénalisation, non plus uniquement à l'instant t_0 de l'assimilation, mais à plusieurs instants t_1, t_2... On se rapproche alors de la formulation à contraintes faibles du 4DVar qui comporte l'inconvénient de l'augmentation de la taille du vecteur de contrôle et donc des coûts de calcul. Mais nous pouvons nous inspirer directement des solutions développées pour palier ces écueils (deux en sont mentionnées à la section (1.2.2)) et proposer par exemple de faire porter la pénalisation exclusivement sur un biais systématique entre l'état à l'instant t_i ayant perdu son caractère parcimonieux à cause de l'erreur modèle et l'état à l'instant t_i soumis à un modèle « parfait » n'impliquant aucune diffusion dans le temps.

Le choix des paramètres λ et p pour bénéficier des avantages numériques de la régularisation (s'ils ne sont pas donnés par la modélisation statistique) a été fait via la résolution de plusieurs problèmes de minimisation, ce qui peut devenir trop coûteux en pratique. Nous pensons que la prochaine priorité devrait être la réduction des ressources nécessaires à la détermination de ces paramètres. Une première idée serait de déterminer dynamiquement ces paramètres au lieu de chercher à les fixer en amont : en mesurant adéquatement leur impact d'une itération à l'autre nous pourrions mettre au point un procédé pour les déterminer à la volée (par un algorithme de bissection par exemple). On pourrait également envisager de se tourner vers les stratégies employées en Machine Learning pour la détermination d'hyperparamètres. Nous pouvons par exemple diviser nos données en un ensemble d'apprentissage et un ensemble de validation, dont la taille permettrait des temps de calcul réalistes pour la mise au point de ces paramètres.

Ensuite, les algorithmes basés sur le gradient conjugué non linéaire et utilisant l'espace dual convergent théoriquement grâce à une technique de « restart » de β_k. Il serait intéressant d'étendre l'étude de la convergence lorsque qu'aucun restart n'est effectué, de même que Al-Baali a étudié la convergence de l'algorithme du gradient conjugué non linéaire sans restart [92]. Cette recherche pourrait amener à mieux gérer l'utilisation des restarts mais est rendue complexe par la non linéarité de l'opérateur de dualité.

Les convergences des algorithmes avec recherche de pas ont été démontrées dans le cas d'utilisation des conditions de Wolfe modifiées. La seconde de ces deux conditions est très chère à évaluer, et il apparaît en pratique suffisant de considérer la première condition (condition d'« Armijo modifiée ») uniquement. Nous savons alors que l'on converge vers un point critique de $f \circ J_q$ mais une preuve de convergence vers un point critique de f dans ce cadre est encore manquante et serait rassurante, notamment pour un cadre d'applications en dehors de l'assimilation de données. En effet, pour cette dernière, nous ne nous attendons de toutes façons pas (à cause du faible nombre d'itérations possibles) à trouver un point critique, que ce soit de f ou de $f \circ J_q$, au sens usuel i.e. une norme relative du gradient de l'ordre de $10^{-8}, 10^{-10}$...

Finalement ces algorithmes ont été pensés pour minimiser un moindre carré non linéaire pénalisé par une norme L_p. Ils s'appliquent également lorsque la norme L_p s'applique au terme de moindre carrés lui-même. Ce cas pourrait être justifié lorsque l'erreur sur les observations suivraient une loi normale généralisée. La nécessité d'utiliser d'autres lois qu'une loi gaussienne pour modéliser les observations est soulignée par Fowler et Van Leeuwen [93]. La loi normale généralisée deviendrait alors pertinente notamment grâce à son paramètre de forme (la variable p dans nos notations) qui permet de ne pas sous-estimer la probabilité d'innovations plus extrêmes.

Troisième partie

Annexes

Annexe A

Comportement de l'algorithme d'Estatico *et al.* avec une recherche linéaire du pas

Nous avons mentionné que dans l'article initial d'Estatico *et al.* [73], le pas devait appartenir à un intervalle très précis (voir Algorithme 6) ce qui permettait d'obtenir la convergence de l'algorithme. La condition que ce pas soit dans cet intervalle est néanmoins une condition suffisante mais pas forcément nécessaire pour la convergence. Nous donnons ici quelques résultats expérimentaux pour montrer que, sur un problème d'assimilation, cette approche fournit des pas trop petits empêchant de converger en pratique. Mais de de meilleurs résultats ont été obtenus avec une recherche linéaire du pas, bien que nous ignorons alors si l'algorithme converge théoriquement ou non.

Nous reprenons le problème d'advection linéaire (sans pénalisation) du chapitre (3), et nous fixons les paramètres $\tau = 11$, $C = 0.9$ et $d = \left((1-\gamma)(1-\tau^{-1}) - \frac{(2+\tau^{-1})^r - 1}{r}\gamma\right)\frac{1}{\|A\|} - 10^{-6}$ d'après les valeurs proposées par l'article. La Table (A.1) compare sur les trois premières itérations l'évolution du résidu $R_n = \|Ax - b\|_2$, la borne maximale T_n de l'intervalle sur lequel est calculé le pas, et le pas α_n.

TABLE A.1 – Étude de la longueur du pas pour l'algorithme d'Estatico *et al.* sur le problème d'advection sans recherche linéaire.

variable \ n	1	2	3
R_n	1330	135.5	135.5
T_n	1.43×10^{-6}	9.50×10^{-8}	1.01×10^{-7}
α_n	1.43×10^{-6}	9.50×10^{-8}	1.01×10^{-7}

Dès la seconde itération le résidu cesse de décroître. Nous remarquons en même temps la décroissance de la valeur de T_n et le pas α_n qui est égal à chaque itération à cette borne maximale. Cela suggère que le pas pourrait être plus grand pour faire décroître la fonctionnelle. Cette hypothèse est appuyée par les résultats de la Table (A.2) qui montre les résultats de la même expérience mais avec une stratégie de backtracking pour le calcul de α_n visant à faire décroître $g(\alpha) = \|AJ_q(x_n^* + \alpha p_n^*) - b\|$. Effectivement, le pas y est plus grand et permet de continuer à faire décroître R_n au fil des itérations.

TABLE A.2 – Étude de la longueur du pas pour l'algorithme d'Estatico sur le problème d'advection avec recherche linéaire

variable \ n	1	2	3	4	5
R_n	1330	125.9	124.1	106.1	74.70
α_n	6.10×10^{-5}	3.05×10^{-5}	3.05×10^{-5}	1.53×10^{-5}	1.53×10^{-5}

Les résultats sur l'expérience shallow-water 2D (5.2.3) amenaient aux mêmes observations : lorsque α_n était contraint à l'intervalle $[0; T_n]$, les itérés n'évoluaient presque pas. L'utilisation de la nouvelle condition de Zoutendijk proposée pourrait alors s'avérer utile pour l'étude de la convergence de l'algorithme avec backtracking (dont les itérations s'effectuent dans l'espace dual), mais cette piste n'a pas encore été explorée.

Annexe B

Méthode proximale et régularisation implicite des algorithmes duaux

Dans cette annexe, nous commençons par décrire brièvement le principe des méthodes proximales, et nous en tirerons un lien avec la régularisation implicite observée au cours des itérations des algorithmes duaux. Ces méthodes sont notamment utilisées pour la minimisation de la somme de deux fonctions $f+g$, lorsque g est un terme de pénalisation en norme L_1 (mais toute autre fonction dont l'opérateur proximal est simple à calculer convient également).

B.1 Méthode proximale

Soit une fonction f convexe, différentiable et lipschitzienne. Nous introduisons l'opérateur proximal de f :

$$\begin{aligned}\text{prox}_f(x_0) &= \underset{x\in\mathbb{R}^n}{\text{argmin}}\ f(x) + \frac{1}{2}\|x-x_0\|^2 \\ &= (Id+\partial f)^{-1}(x_0)\end{aligned}$$

qui est bien défini car $f(x)+\frac{1}{2}\|x-x_0\|^2$ est strictement convexe et coercif.

Nous avons la caractérisation suivante : x est un minimum de f si et seulement c'est un point fixe de prox_f : $x \in \underset{x\in\mathbb{R}^n}{\text{argmin}} f(x) \iff \text{prox}_f(x) = x$. Or, sous ces hypothèses, prox_f (aussi appelé résolvante de f) est un opérateur contractant [96]. On peut alors minimiser f grâce aux itérations

$$x_{n+1} = \text{prox}_{\alpha_n f}(x_n) \tag{B.1}$$

où $(\alpha_n)_n$ est une suite de réels positifs.

L'opérateur proximal est également utilisé pour la minimisation de la somme de deux fonctions $\min f + g$ avec f comme ci-dessus et g semi-continue inférieurement, convexe et propre (abrégée fonction CCP pour Closed Convex Proper) en utilisant le schéma itératif

$$x_{k+1} = \mathrm{prox}_{\lambda g}(x_k - \tau \nabla f(x_k)). \tag{B.2}$$

Pour $g = \|\cdot\|_1$, l'opérateur proximal correspond au « seuillage doux » :

$$\mathrm{prox}_{\lambda\|\cdot\|_1}(x) = \mathrm{Signe}(x)\max(|x| - \lambda, 0) \tag{B.3}$$

qui vient mettre à zéro les composantes de x_{k+1} appartenant à $[-\lambda; \lambda]$. Nous ajoutons souvent à cette méthode un procédé d'inertie qui donne du poids aux « bonnes » itérations qui ont permis de progresser le plus : nous parlons d'accélération de Nesterov qui donne lieu à l'algorithme FISTA (Fast Iterative Soft Thresholding Algorithm [94]).

Qu'en est-il du choix de $g = \|\cdot\|_p^p$? En utilisant la définition de l'opérateur proximal, nous avons pour tout $\tau > 0$ et pour tout $x = (x_1, \ldots, x_n) \in \mathbb{R}^n$:

$$\mathrm{prox}_{\tau\|\cdot\|_p^p}(x) = \begin{pmatrix} \mathrm{prox}_{\tau|\cdot|^p}(x_1) \\ \vdots \\ \mathrm{prox}_{\tau|\cdot|^p}(x_n) \end{pmatrix} \tag{B.4}$$

Cependant l'expression de $\mathrm{prox}_{\tau|\cdot|^p}(x_i)$ n'a de forme analytique que pour certaines valeurs de p. L'article [43] fournit de telles formes analytiques pour $p = \frac{1}{2}$ et $p = \frac{2}{3}$ et fournit ainsi qu'un algorithme du calcul de $\mathrm{prox}_{\tau|\cdot|^p}$ pour $0 < p < 1$. La complexité de l'algorithme proximal dans le cas $g = \|\cdot\|_p^p$ dépend alors de ce dernier calcul. De plus, le cas $p > 1$ n'est pas traité.

B.2 Régularisation implicite des méthodes duales

Illustration graphique

Les expériences sur les équations de Navier-Stokes ont montré que NLCG parvenait à une solution visiblement régularisée même pour un paramètre de régularisation nul $\lambda = 0$. Nous allons constater que ce phénomène est partagé par d'autres algorithmes dont les itérations s'effectuent dans le dual en affichant l'état au cours des itérations, avant d'en donner une explication intuitive. Nous reprenons $p = 1.2$ pour l'ensemble de cette section.

Pour mieux observer cette régularisation, nous reprenons l'exemple simple de l'advection unidimensionnelle. Le problème aux moindres carrés sans pénalisation à minimiser possède une solution affichée sur la Figure (B.1). La Figure (B.2) montre quelques itérations (de numéro 10, 100 et 300) pour les algorithmes de descente de gradient (B.2a) et de descente de gradient dans l'espace dual (B.2b). Nous observons effectivement que les itérations de la descente de gradient classique sont plus lisses et plus oscillantes que celles de la descente de gradient dans le dual qui sont, elles, plus abruptes.

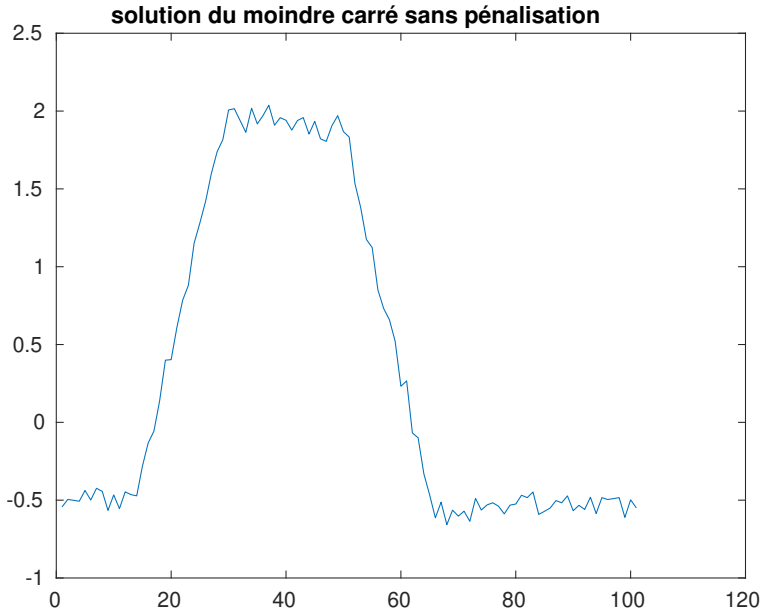
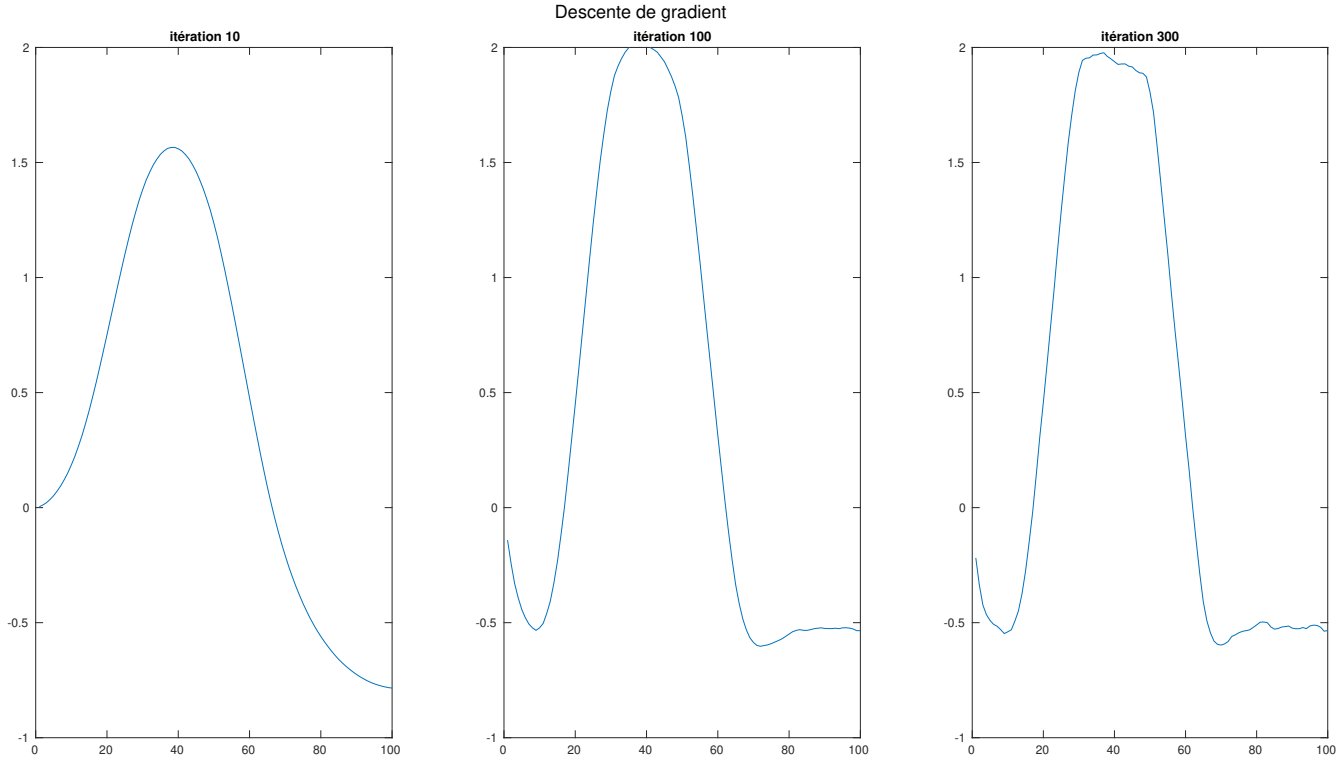


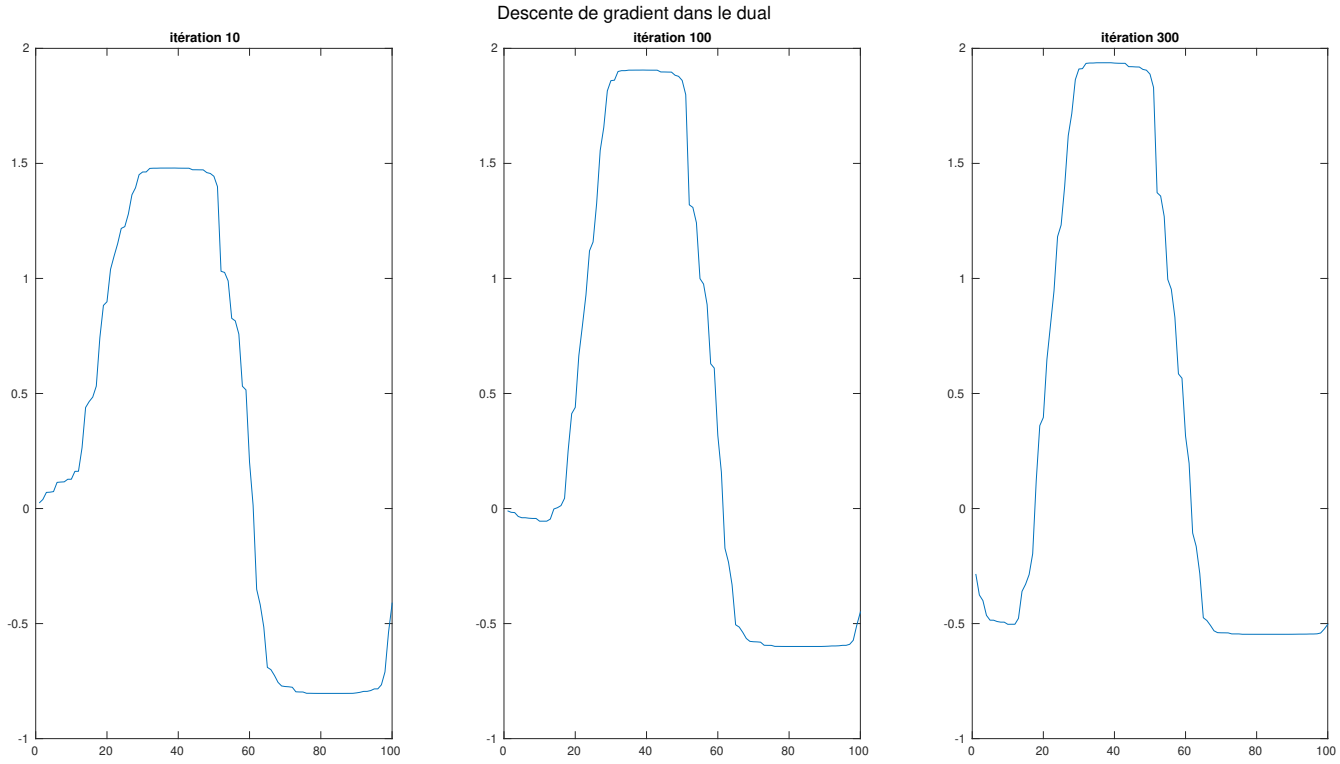
FIGURE B.1 – Solution du problème aux moindre carrés sans pénalisation.

De même la Figure B.2 montre les itérations numéro 5, 10 et 15 des algorithmes d'ordre 2. Les itérations du gradient conjugué non linéaire servent ici de témoins sur la Figure (B.3a). Similairement à la descente de gradient classique nous retrouvons des itérations oscillantes, tandis que la descente de gradient dans l'espace dual de la Figure (B.3b) et l'algorithme d'Estatico *et al.* de la Figure (B.3c) partagent des itérations parcimonieuses. Ces dernières n'ont pas encore été affectées par le bruit qui caractérise la vraie solution du système.

Les algorithmes duaux semblent donc prendre un chemin vers la solution tel qu'il l'aurait été si la fonctionnelle eusse été pénalisée. Une interruption prématurée des itérations comme le demande l'assimilation de données impose donc une régularisation implicite particulière aux algorithme duaux.

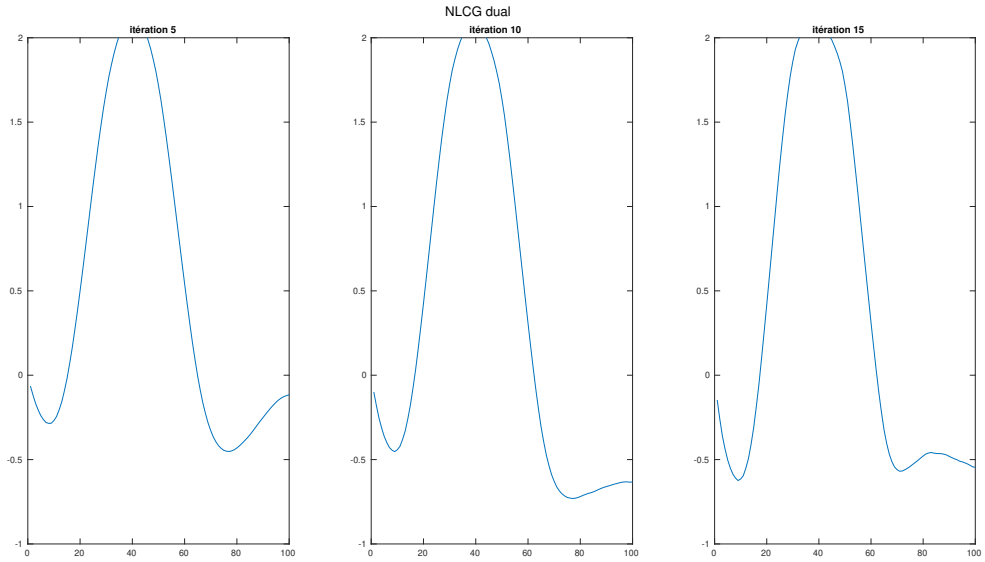


(a) Affichage des itérations de la descente de gradient.

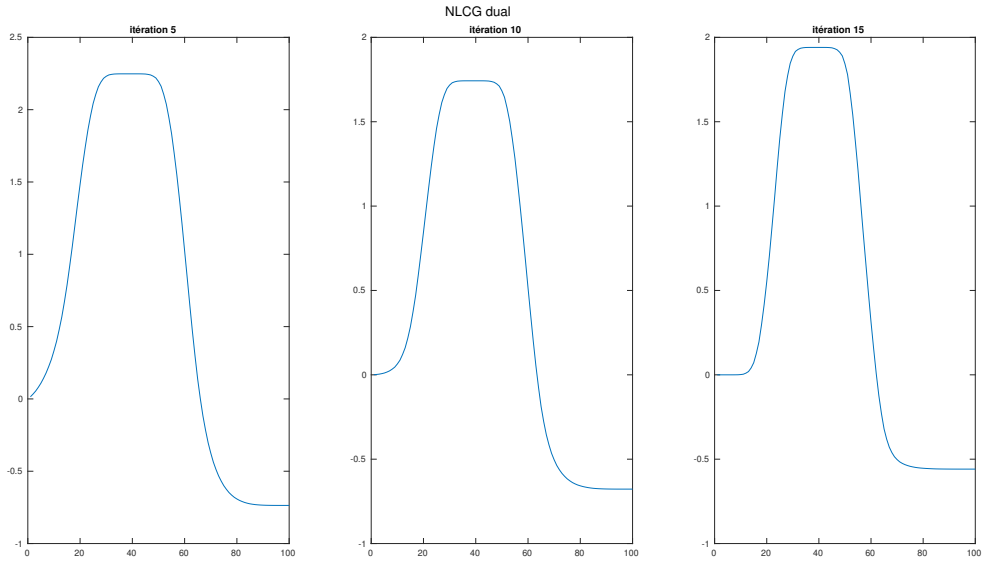


(b) Affichage des itérations de la descente de gradient dans le dual.

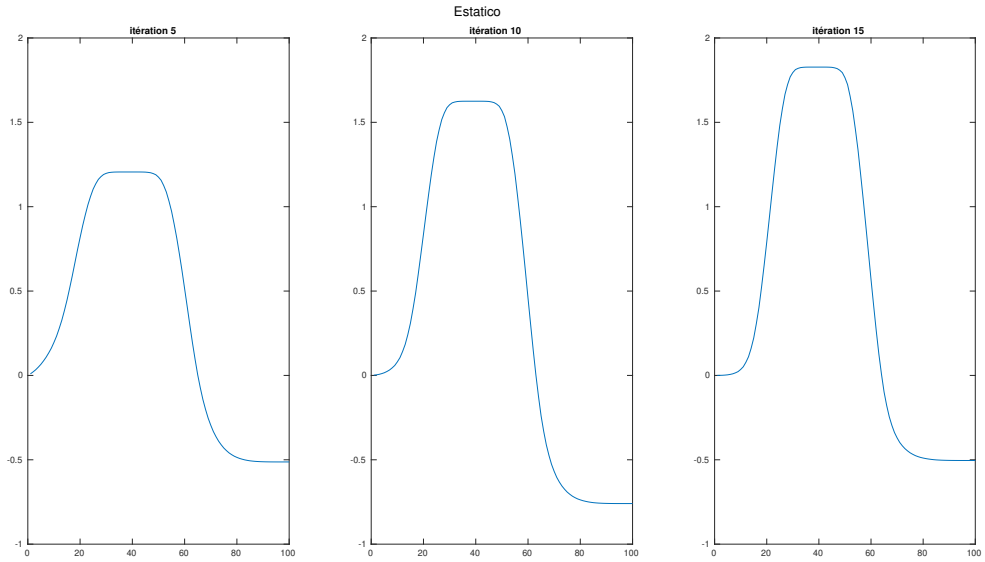
FIGURE B.2 – Régularisation implicite au cours des itérations pour les algorithmes d'ordre 1.



(a) Affichage des itérations de NLCG.



(b) Affichage des itérations de NLCG dual.



(c) Affichage des itérations de l'algorithme d'Estatico.

FIGURE B.3 – Régularisation implicite au cours des itérations pour les algorithmes d'ordre 2.

Explication intuitive

Réécrivons le schéma itératif d'un algorithme dual :

$$x_{k+1} = J_q(J_p(x_k) - \alpha_k \nabla f_k). \tag{B.5}$$

et comparons le au Iterative Soft Thresholding Algorithm (correspondant aux itérations B.2 utilisant l'équation B.3) qui effectue un seuillage doux des itérations. Ici, ce n'est plus l'opérateur proximal de la norme L_1 qui est appliqué à gauche mais l'opérateur de dualité J_q. Ce changement correspond à un type de seuillage différent comme le montre la Figure (B.4) sur lequel nous affichons le seuillage doux et $J_q(x)$ en une dimension avec $q = 6$, ce qui correspond à $p = 1.2$. De son côté, l'étape (B.5) correspond également à un seuillage illustré sur la Figure (B.4).

Si x appartient à $\mathbb{R}^n$ ce seuillage s'effectue composante par composante (comme pour le seuillage doux) en vertu du fait que $J_q(x) = (J_q(x_1), \ldots, J_q(x_n))$. D'après la Figure (B.4), les petites composantes des itérations sont donc mises à zéro mais le seuillage se fait de façon plus lisse que dans le cas de l'équation (B.3).

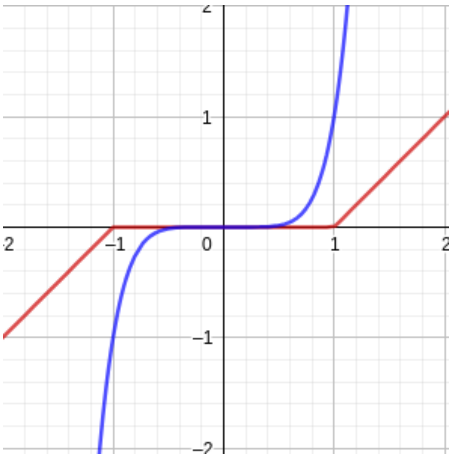


FIGURE B.4 – Seuillage doux $y(x) = \text{Signe}(x)\max(|x| - \lambda, 0)$ (en rouge) et seuillage effectué par l'opérateur de dualité $y(x) = \text{Signe}(x)|x|^q$ avec $q = 6$ soit $p = 1.2$ (en bleu).

Annexe C

Algorithmes duaux et descente miroir

Nous montrons ici que nous pouvons réinterpréter les descentes de gradient duales proposées comme des « descentes miroirs » (mirror descent algorithm) que nous présentons en même temps.

Ce type de descente est une généralisation de l'algorithme de descente de gradient capable de s'adapter à la géométrie du problème. Nous retrouvons ses itérations en remarquant que la descente de gradient classique est équivalente à chercher le minimum de la fonction

$$\mathcal{F}(x) = f_k + \alpha_k \langle \nabla f_k, x \rangle + \frac{1}{2}\|x - x_k\|_2^2.$$

En effet, prendre la condition d'optimalité du premier ordre donne $x_{k+1} = x_k - \alpha_k \nabla f_k$. Autrement dit, nous cherchons à minimiser le développement à l'ordre 1 de f avec une pénalisation qui nous empêche de partir trop loin de x_k, où cette approximation ne serait plus valable.

Plus généralement, nous pouvons considérer un terme de pénalisation différent $D(x, x_k)$. En prenant $D(x, x_k) = \frac{1}{p}\|x - x_k\|_p^p$ nous obtenons la mise à jour de la descente de gradient avec transport de la direction dans le primal :

$$\begin{aligned} \nabla \mathcal{F}(x_{k+1}) = 0 &\iff \alpha_k \nabla f_k + J_p(x_{k+1} - x_k) = 0 \\ &\iff x_{k+1} = x_k - J_q(\alpha_k \nabla f_k). \end{aligned}$$

Un autre cas particulier très utilisé en pratique, introduit dans [97], utilise pour D la distance de Bregman associée à une fonction strictement convexe et différentiable h :

$\Delta_h(x,y) = h(y) - h(x) - \langle \nabla h(x), y - x \rangle$. L'itération de la descente de miroir dans ce cas est

$$x_{k+1} = \nabla h^*(\nabla h - \alpha_k \nabla f_k). \tag{C.1}$$

avec g^* la transformée de Legendre-Fenchel de g : $g^*(y) = \sup_{x \in \mathbb{R}^n} (\langle x, y \rangle - g(x))$. Comme pour toute fonction g CCP nous avons $(\partial g)^{-1} = \partial(g^*)$, nous obtenons $\partial \left[\left(\frac{1}{p} \| \cdot \|_p^p \right)^* \right] = \left[\partial \left(\frac{1}{p} \| \cdot \|_p^p \right) \right]^{-1} = J_q$. Ainsi l'itération de la descente de gradient dans l'espace dual

$$x_{k+1}^* = x_k^* - \alpha_k \nabla f_k \iff x_{k+1} = J_q(J_p(x_k) - \alpha_k \nabla f_k).$$

est équivalente à la mise à jour (C.1) en prenant $h = \frac{1}{p} \| \cdot \|_p^p$.

Bibliographie

[1] S.L. Barnes, *A technique for maximizing details in numerical weather map analysis,* J. Appl. Meteor. 3 (1963) 396–409. 11

[2]] L.S. Gandin, *Objective analysis of meteorological fields,* Translated from Russian by the Israeli Program for Scientific Translations, 1965. 11

[3] Asch, Mark et Bocquet, Marc et Nodet, Maëlle, *Data Assimilation, Methods, Algorithms and Applications,* Society for Industrial and Applied Mathematics, 2016. 11

[4] A. C. Lorenc, *Analysis methods for numerical weather prediction,* Q. J. R. Meteorol. Soc., 112 :1177–1194, 1986. 17

[5] K. Ide et P. Courtier et M. Ghil, et A. C. Lorenc, *Unified notation for data assimilation,* Operational, sequential and variational. J. Met. Soc. Japan, 75, 1997. 15

[6] R. Kalman, *A new approach to linear filtering and prediction problems,* Journal of Physical Oceanography, 23 :2541–2566, 1960. 18

[7] L. Bertino, G. Evensen, et H. Wackernagel, *Sequential data assimilation techniques in oceanography,* International Statistical Review, 71 :223–242, 2003. 18

[8] G. Evensen, *Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics* Journal of Geophysical Research, 99 :143-162, 1994. 22

[9] G. Evensen, *Data Assimilation : The Ensemble Kalman Filter,* Springer, 2009. 22

[10] Sakov, P. et Oke, P.R., *A deterministic formulation of the ensemble Kalman filter : an alternative to ensemble square root filters,* Tellus A, 60 : 361-371, 2008. 23

[11] Geir Evensen et Peter Jan Van Leeuwen et Femke C. Vossepoel, *Data Assimilation Fundamentals, A Unified Formulation of the State and Parameter Estimation Problem,* Springer, 2022. 18

[12] A.H. Jazwinski, *Stochastic Processes and Filtering Theory,* Academic Press, New York, 1970. 20

[13] M. Verlaan et A.W. Heemink, *Tidal flow forecasting using reduced rank square root filters,* Stoch. Hydrol. Hydraul., 11(5) :349–368, 1997. 22

[14] D.T. Pham, J. Verron, et M.-C. Roubaud, *A singular evolutive extended kalman filter for data assimilation in oceanography,* Journal of Marine Systems, 16,3-4 :323–340, 1998. 22

[15] M.K. Tippett, J.L. Anderson, C.H. Bishop, T.M. Hamil, J. S Whitaker, *Ensemble square root filters,* Mon., Wea. Rev. 131 (2003) 1485–1490. 23

[16] P.L. Houtekamer, H.L. Mitchell, *A sequential ensemble Kalman filter for atmospheric data assimilation,* Mon. Wea. Rev. 129 (2001) 123–137. 23

[17] J.L. Anderson, S.L. Anderson, *A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts,* Mon. Wea. Rev. 127 (1999) 2741–2758. 23

[18] J.S. Whitaker, T.M. Hamil, *Ensemble data assimilation without perturbed observations,* Mon. Wea. Rev. 130 (2002) 1913–1924. 23

[19] J.S. Whitaker, T.M. Hamil, *A fundamental study of the numerical prediction based on the variational principle,* J. Met. Soc. Japan, 33 :262–265, 1955. 11, 24

[20] P. Courtier, J. N. Thépaut, et A. Hollingsworth, *A strategy for operational implementation of 4d-var, using an incremental approach,* Q. J. R. Meteorol. Soc., 120 :1367–1387, 1994. 25

[21] S. Gratton et A.S. Lawless et N.K. Nichols, *Approximate Gauss-Newton methods for nonlinear least squares problems,* SIAM J. Optim., 18(1) : 106-132, 2007. 45

[22] W.F. Mascarenhas, *The divergence of the BFGS and Gauss Newton Methods,* Mathematical Programming, 147 : 253–276, 2014. 45

[23] J.L. Lions, *Contrôle optimal des systèmes gouvernés par des équations aux dérivées partielles,* Dunod, Paris, 1968. 28

[24] F.-X. Le Dimet, *A general formalism of variational analysis,* CIMMS Report, 1982. 28

[25] F.-X. Le Dimet et O. Talagrand, *Variational algorithms for analysis et assimilation of meteorological observations : Theorotical aspects,* Tellus, 38A :97–110, 1986. 28

[26] Blayo, E., Bocquet, M., Cosme, E., Cugliandolo, L. F., *Advanced Data Assimilation for Geosciences,* Oxford University Press, lecture Notes of the Les Houches School of Physics : Special Issue, June 2012. 29

[27] A.S. Lawless, N.K. Nichols, S.P. Ballard, *A comparison of two methods for developing the linearization of a shallow-water model,* Q. J. R. Meteor. Soc. 129 (2003) 1237–1254. 29

[28] Tonani, M., M. Balmaseda, L. Bertino, E. Blockley, G. Brassington, F. Davidson, Y. Drillet, et al., *Status and Future of Global and Regional Ocean Prediction Systems,* Journal of Operational Oceanography 8 (s2) : s201–s220. 30

[29] P. Courtier, *Dual formulation of four-dimensional variational assimilation,* Q. J. R. Meteor. Soc. 123 (1997) 2449-2461. 30

[30] J. Nocedal, S. Wright S, *Numerical optimization,* Springer . 1999. 30, 42, 46, 111

[31] Björck Å, *Numerical methods for least squares problems,* SIAM : Philadelphia, USA. 1996. 32

[32] Mallat S, *A Wavelet Tour of Signal Processing,* Academic Press ; 3rd edition . 2008. 32, 121

[33] Freitag, M., N. Nichols, et C. Budd, *Resolution of sharp fronts in the presence of model error in variational data assimilation,* Quartely journal of the royal meteorological society, 139, 742–757. 2013. 12, 32, 33, 65, 69

[34] Michelle M. Gierach et Bulusu Subrahmanyam et Annette Samuelsen et Kyozo Ueyoshi, *Hurricane-driven alteration in plankton community size structure in the Gulf of Mexico : A modeling study,* Geophysical Reasearch Letters, Vol. 36, 2009.

[35] Hui Zou et Trevor Hastie, *Regularization and variable selection via the elastic net,* Journal of the Royal Statistical Society : Series B (Statistical Methodology), vol. 67, no 2, 2005, p. 301-320.

[36] Peng Zhao et Bin Yu, *On Model Selection Consistency of Lasso,* The Journal of Machine Learning Research, vol. 7, 2006, p. 2541-2563 34

[37] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu et Keith Knight, *Sparsity and smoothness via the fused lasso,* Journal of the Royal Statistical Society : Series B (Statistical Methodology), vol. 67, no 1, 2005, p. 91-108. 34

[38] Li, Xudong et Sun, Defeng et Toh, Kim-Chuan, *A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems,* doi 10.48550/ARXIV.1607.05428, arXiv, 2016. 35

[39] Huber, Peter J., *Robust Estimation of a Location Parameter,* Annals of Statistics. 53 (1) : 73–101, (1964). 35

[40] A. Alexanderian, N. Petra, G. Stadler, et O. Ghattas, *A-optimal design of experiments for infinite-dimensional Bayesian linear inverse problems with regularized 0-sparsification,* SIAM J. Sci. Comput., 36 (2014), pp. A2122–A2148. 35

[41] Louizos, Christos et Welling, Max et Kingma, Diederik P., *Learning Sparse Neural Networks through L_0 Regularization,* arXiv, 2017. 35

[42] Bo Jiang, Ya-Feng Liu, Zaiwen Wen, *L_p-norm regularization algorithms for optimization over permutation matrices,* SIAM Journal on Optimization, Vol. 26, Iss. 4, p. 2284-2313, 2016. 35

[43] Chen, F., Shen, L. et Suter, B.W, *Computing the proximity operator of the lp norm with $0 < p < 1$,* IET Signal Process., 10 : 557-565, 2016. 35, 146

[44] Chen, Xiaojun et Xu, Fengmin et Ye, Yinyu, *Lower Bound Theory of Nonzero Entries in Solutions of ℓ_2-ℓ_p Minimization,* SIAM Journal on Scientific Computing. 32. 2010. 35

[45] F. Lenti et F. Nunziata et C. Estatico et M.Migliaccio, *Analysis of reconstructions obtained solving l^p-penalized minimization problems,* IEEE Trans. Geosci. Remote Sens, Vol. 53, p. 4876-4886, 2015. 35

[46] Wang, Y., I. Navon, X. Wang, et Y. Cheng, *D Burgers equation with large Reynolds number using POD/DEIM and calibration,* International Journal for Numerical Methods in Fluids, 82 (12), 909–931, 2016. 37

[47] Anzengruber, Stephan et Ramlau, Ronny, *Morozov's discrepancy principle for Tikhonov-type functionals with nonlinear operators,* Inverse Problems. 26. 025001. 10.1088/0266-5611/26/2/025001 (2009). 37

[48] T. Schuster et B. Kaltenbacher et B. Hofmann et K.S. Kazimierski, *Regularization method in Banach spaces,* Radon series on Computational and applied mathematics, 2012. 35, 48, 49

[49] B.P. Rynne et M.A. Youngson, *Linear Functional Analysis,* Springer, SUMS, 2008. 51

[50] W. Schirotzek, *Nonsmooth Analysis,* Springer, Berlin, 2007. 51

[51] J. Lindenstrauß et L. Tzafriri, *Classical Banach Spaces, II.* Springer, Berlin, 1979. 52

[52] Z.-B. Xu et G. F. Roach, *Characteristic inequalities of uniformly convex and uniformly smooth Banach spaces.* Journal of Mathematical Analysis and Applications, 157(1) :189–210, 1991. 53, 56, 58

[53] Saad, Y. *Iterative methods for sparse linear systems. PWS* Publishing Company : Boston, USA, 1996. 45, 47

[54] Levenberg K., *A method for the solution of certain problems in least squares,* Quarterly Journal on Applied Mathematics 2 : 164–168, 1944. 45

[55] Marquardt D., *An algorithm for least-squares estimation of nonlinear parameters,* SIAM Journal on Applied Mathematics 11 : 431–441, 1963. 45

[56] Conn AR, Gould NIM, Toint PL., *Trust-region methods,* No. 01 in : MPS-SIAM Series on Optimization, SIAM : Philadelphia, USA, 2000. 45

[57] Engeli M, Ginsburg T, Rutishauser H, Stiefel E., *Refined iterative methods for computation of the solution and the eigenvalues of self-adjoint bondary value problems,* (mitteilungen aus dem institut für angewandte mathematik der eth zürich, nr. 8.) 107 s. basel/stuttgart 1959. birkhäuser verlag. preis brosch. dm 17,. ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik 40(10-11) : 525–525, doi :10.1002/zamm.19600401041, 1960. 46

[58] Kaniel S., *Estimates for some computational techniques in linear algebra,* Mathematics of Computation 20(95) : 369–378, 1966. 46

[59] Gratton, S. et Tshimanga, J., *An observation-space formulation of variational assimilation using a restricted preconditioned conjugate gradient algorithm,* Q.J.R. Meteorol. Soc., 135 : 1573-1585, 2009. 46, 83, 123

[60] W. Hager et Hongchao Zhang, *A survey of nonlinear conjugate gradient methods,* Pacific Journal of Optimization, 2 : 35-58, 2006. 47

[61] C. G. Broyden, *The Convergence of a Class of Double-rank Minimization Algorithms,* Journal of the Institute of Mathematics and Its Applications, vol. 6, 1970, p. 76-90. 48

[62] R. Fletcher, *A New Approach to Variable Metric Algorithms,* Computer Journal, vol. 13, 1970, p. 317-322. 48

[63] D. Goldfarb, *A Family of Variable Metric Updates Derived by Variational Means,* Mathematics of Computation, vol. 24, 1970, p. 23-26. 48

[64] D. F. Shanno, *Conditioning of Quasi-Newton Methods for Function Minimization,* Mathematics of Computation, vol. 24, 1970, p. 647-656. 48

[65] Y. Sasaki, *Some basic formalisms in numerical variational analysis,* Mon. Wea. Rev. 98 (1970) 875–883. 30

[66] Y. Trémolet, *Accounting for an imperfect model in 4D-Var,* Q. J. R. Meteor. Soc. 132 (2006) 2483–2504. 31

[67] J.C. Gilbert et J. Nocedal, *Global Convergence Properties of Conjugate Gradient Methods for Optimization,* SIAM J. Optim. 2 (1992) 21-42. 43, 106

[68] J. Nocedal, *Updating Quasi-Newton matrices with limited storage,* Math. Comput. 38A (1986) 137–161. 48

[69] Lanczos C., *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators,* Journal of research of the National Bureau of Standards B 45 : 225–280, 1950. 47

[70] F Schöpfer et A. K. Louis et T. Schuster, *Nonlinear iterative methods for linear ill-posed problems in Banach spaces*, Inverse Problems 22 311. 54

[71] T. Bonesky et K.S. Kazimierski et P. Maass et F. Schöpfer et T. Schuster, *Minimization of Tikhonov functional in Banach spaces,* Abstract and applied analysis 2008 : 1085-3375, 2008. 56, 87

[72] Herzog, Roland et Wollner, Winnifried, *A conjugate direction method for linear systems in Banach spaces* Journal of Inverse and Ill-posed Problems, vol. 25, no. 5, 2017, pp. 553-572. https ://doi.org/10.1515/jiip-2016-0027. 57

[73] Estatico, Claudio et Gratton, Serge et Lenti, Flavia et Titley-Peloquin, *A conjugate gradient like method for p-norm minimization in functional spaces,* Numerische Mathematik, 137 (4). 895-922. ISSN 0029-599X. 58, 143

[74] David W. Boyd, *The power method for lp norms, Linear Algebra and its Applications,* vol. 9, pp. 95-101 (1974). 59

[75] Higham, N.J. *Estimating the matrixp-norm,* Numer. Math. 62, 539–555 (1992). 59

[76] A. Bernigaud et S. Gratton et F. Lenti et E. Simon et O. Sohab Lp-norm regularization in variational data assimilation, *Q. J. R. Meteorol. Soc.*, 147, 2067-2081, 2021. 61

[77] Nazanin Asadi et K. Andrea Scott et David A. Clausi, *Data fusion and data assimilation of ice thickness observations using a regularisation framework,* Tellus A : Dynamic Meteorology and Oceanography, Volume 71, 2019 - Issue 1. 61, 63

[78] Dytso, A., Bustin, R., Poor, H. et al. *Analytical properties of generalized Gaussian distributions,* J Stat Distrib App 5, 6 (2018). https ://doi.org/10.1186/s40488-018-0088-5. 61

[79] Goodman, I. R. et Kotz, S. *Multivariate θ-generalized normal distributions,* J. Multivar. Anal. 3(2), 204–219 (1973). 62, 64

[80] De Simoni, S. *Su una estensione dello schema delle curve normali di ordine r alle variabili doppie. Statistica,* 37, 447–474 (1968) 63

[81] A. A. Roenko, V. V. Lukin, I. Djurović et M. Simeunović, *Estimation of parameters for generalized Gaussian distribution,* 6th International Symposium on Communications, Control and Signal Processing (ISCCSP), 2014, pp. 376-379, doi : 10.1109/ISCCSP.2014.6877892. 64

[82] Frédéric Pascal, Lionel Bombrun, Jean-yves Tourneret, yannick Berthoumieu, *Parameter Estimation For Multivariate Generalized Gaussian Distributions,* IEEE Transactions on Signal Processing, Institute of Electrical and Electronics Engineers, 2013, 61 (23), pp.5960-5971. 64

[83] Banerjee, S. et Agrawal, M., *Underwater acoustic noise with generalized Gaussian statistics : Effects on error performance,* Proceedings of OCEANS - Bergen, 2013 MTS/IEEE, pp. 1–8. IEEE, Bergen, (2013). 12, 64

[84] Julien Jouanno, Rachid Benshila, Léo Berline, Antonin Soulié, Marie-Hélène Radenac, et al., *A NEMO-based model of Sargassum distribution in the tropical Atlantic : description of the model and sensitivity analysis (NEMO-Sarg1.0.),* Geoscientific Model Development Discussions, Copernicus Publ, 2021, 14 (6), pp.4069 - 4086. ⟨10.5194/gmd-14-4069-2021⟩. ⟨hal-03287064⟩ 65

[85] Peter Lax et Burton Wendroff *Systems of conservations laws,* Commun. Pure Appl. Math., 1960, 13, pp. 218-237. 66

[86] J. Pedloski, *Geophysical Fluid Dynamics,* Springer, 1986. 116

[87] R. Asselin, *Frequency Filter for Time Integrations,* Monthly Weather Review, 1972, 100, pp. 487-490. 116

[88] Yong Li, Catalin Trenchea *Analysis of time filters used with the leapfrog scheme,* 2015. 116

[89] R. Daley, *Atmospheric data analysis,* Cambridge University Press, 1991. 118

[90] I. Mirouze et A. Weaver, *Representation of correlation functions in variational assimilation using an implicit diffusion operator,* Quarterly Journal of the Royal Meteorological Society, 654, pp. 1421 - 1443, 1991. 118

[91] Gratton S., Toint P.L., Tröltzsch A., *An active-set trust-region method for derivative-free nonlinear bound-constrained optimization,* Optim. Methods Softw., 26, 873-894, (2011). 122

[92] M. Al-Baali, *Descent property and global convergence of the Fletcher-Reeves method with inexact line search,* I.M.A. Journal on Numerical Analysis, 5 (1985), pp. 121–124. 139

[93] Alison Fowler et Peter Jan Van Leeuwen, *Observation impact in data assimilation : the effect of non-Gaussian observation error,* Tellus A : Dynamic Meteorology and Oceanography, Vol. 65, No. 1, 2013. 140

[94] Amir Beck et Marc Teboulle, *A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems,* SIAM J. Vol. 2, No. 1, pp. 183–202, 2009. 146

[95] Safeer Hussain Khan, *Iterative convergence of resolvents of maximal monotone operators perturbed by the duality map in Banach spaces,* Acta Mathematica Academiae Paedagogicae Nyiregyhaziensis, January 2004.

[96] Heinz H. Bauschke et Patrick L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces,* Springer 2011. 145

[97] Amir Beck, Marc Teboulle, *Mirror descent and nonlinear projected subgradient methods for convex optimization,* Operations Research Letters, Volume 31, Issue 3, 2003, Pages 167-175. 151