

UNIVERSITÉ DE NICE-SOPHIA ANTIPOLIS - UFR SCIENCES

École Doctorale

THÈSE

Présentée pour obtenir le titre de

Docteur en SCIENCES
de l'Université de Nice Sophia Antipolis

Spécialité :

MATHÉMATIQUES

par

Didier AUROUX

Étude de différentes méthodes d'assimilation de données
pour l'environnement

Soutenue publiquement le 10 Décembre 2003
devant le jury composé de :

M. Yann BRENIER	CNRS - Université de Nice	Président
M. Alain BENSOUSSAN	Université de Paris Dauphine	Rapporteur
M. Olivier TALAGRAND	CNRS - École Normale Supérieure	Rapporteur
M. Éric BLAYO	Université de Grenoble	Examineur
M. François-Xavier LE DIMET	Université de Grenoble	Examineur
M. Jacques BLUM	Université de Nice	Directeur
M. Mohamed JAOUA	Université de Nice	Membre invité

14 heures 30

Laboratoire J.A. Dieudonné, Parc Valrose, 06108 NICE Cedex 2

Remerciements

Je tiens à remercier en tout premier lieu mon directeur de thèse, Jacques Blum. Je le connais maintenant depuis près de 6 ans. Il a d'abord été pour moi un professeur parmi tant d'autres avant de devenir mon maître de stage en licence, puis en maîtrise. De plus en plus passionné par tout ce qui avait trait à l'assimilation de données, j'ai donc décidé d'en remettre une couche et de faire mon DEA avec lui. C'est alors tout naturellement que je l'ai suivi dans cette contrée lointaine et ensoleillée qu'est la ville de Nice pour faire ma thèse avec lui. Il a énormément contribué à ce que ma thèse se déroule dans d'excellentes conditions scientifiques et dans une bonne ambiance générale due à la proximité géographique de nos bureaux respectifs et à sa bonne humeur inébranlable, même dans certains moments délicats.

Je remercie ensuite l'ensemble des membres du jury, qui m'ont fait l'honneur de bien vouloir étudier avec attention mon travail : Alain Bensoussan et Olivier Talagrand pour avoir accepté d'être rapporteurs de cette thèse et pour m'avoir fourni une multitude de pistes de recherche pour les prochaines années ; Éric Blayo et François-Xavier Le Dimet pour avoir accepté d'examiner cette thèse et pour m'avoir aidé et bien accueilli à Grenoble lors de mes nombreux séjours ; Mohamed Joua pour avoir accepté mon invitation à participer à ce jury ; et enfin Yann Brenier pour m'avoir fait l'honneur d'accepter de présider ce jury.

Mes remerciements s'adressent également aux membres du LMC à Grenoble qui m'ont beaucoup aidé et qui m'ont hébergé pendant mes différents stages chez eux. Cette thèse a d'ailleurs été réalisée dans le cadre du projet IDOPT de l'INRIA Rhône-Alpes et dont le camp de base est établi au LMC.

Je tiens ensuite à exprimer toute ma gratitude envers le personnel *administratif* du JAD, à savoir dans le désordre (mais par ordre alphabétique) Annie, Fernande, Heike, Isabelle, Janine, Jean-Louis, Jean-Paul, Marie-Christine, Marie-Claude et Rosalba, pour leur extrême compétence, disponibilité et gentillesse. Par la même occasion, je remercie les ingénieurs informaticiens Bernard et Jean-Marc qui m'ont laissé faire mes bêtises sur mon ordinateur, et Messieurs les Directeurs successifs du laboratoire

que j'ai vus défiler, Frédéric et Gilles.

Passons ensuite aux joyeux drilles du 4^{ème} : Jacques (en bon chef de file), mes co-bureaux Jean-Pascal et Maëlle avec qui ce fut un réel plaisir de partager le 822, mes voisins proches Georges et Mohamed, les (ex-)thésards du 4^{ème} (ceux qui voulaient prendre le pouvoir et qui sont de vrais amis sur qui on peut compter) : Kro (en version rousse, blonde ou brune, peu importe, elle sait tout ce que je lui dois pour ces 3 années), Maëlle (qui visiblement continue à me supporter puisqu'elle n'a pas encore demandé à changer de bureau), Mat3 (homonyme de la machine, mais heureusement il a planté moins souvent qu'elle!), Alexis 1^{er} du nom (même s'il n'est plus thésard) et sa famille, Guil (sponsorisé par Chocapic et par un canard qui fait coin coin coin ...) et Bianca, Pachou (rame sa chaire, comprenne qui pourra), Jamil (devenu strasbourgeois entre temps), Oliv (recordman du labo), David (de loin le plus suspect du 801), Jeff (même s'il dit régulièrement du mal de l'OL et de l'ASVEL), Emmanuel (un ex-cobureau), Jean-Pascal (un co-bureau actuel), Greg (le rugbyman), et tous les thésards fantômes qu'on a jamais vus. Merci à tous les profs et maîtres de conférence de l'étage qui ont également beaucoup contribué à la renommée internationale du 4^{ème}.

Je n'oublierai pas les exilés des étages inférieurs, Francesca, Amina, Angela, Marie, Mathieu (celui du 3), Baohua, Fabien, Matthieu (celui du 2), Uwe, les exilés lointains comme Jean-Pierre (et sa sœur), et tous ceux que j'ai quand même oubliés.

Merci enfin à mes frères, à mon père, à ma mère, et surtout à ma grand-mère.

Table des matières

1	Introduction	9
2	L'assimilation de données	13
2.1	Qu'est-ce que l'assimilation de données?	13
2.2	Différentes approches	14
2.2.1	Assimilation séquentielle	15
2.2.2	Assimilation variationnelle	20
2.3	Difficultés actuelles et objectifs	28
2.3.1	Difficultés et enjeux	28
2.3.2	Projets opérationnels	29
2.3.3	Objectifs	30
3	Minimisation	31
3.1	Introduction	31
3.2	Algorithmes de descente	32
3.2.1	Algorithme de Newton	32
3.2.2	Algorithme de type quasi-Newton	32
3.2.3	Algorithme BFGS	33
3.2.4	Algorithme L-BFGS	33
3.2.5	Mise à jour de la matrice diagonale	34
3.2.6	Mise à l'échelle de la matrice diagonale	35
3.3	Hessienne et covariances d'erreur	35
3.4	Application à l'équation de Burgers	37
3.4.1	Équations du modèle	37
3.4.2	Paramètres des expériences numériques	37
3.4.3	Convergence de l'algorithme BFGS à mémoire illimitée	38
3.4.4	Choix des paires stockées	38
3.4.5	Impact de la mise à l'échelle	41
3.4.6	Nouvelle approche de la mise à l'échelle	43
3.4.7	Choix de M	43
3.4.8	Évolution pendant la minimisation	46

3.4.9	Cas non quadratique	46
3.4.10	Conclusions relatives à ce problème test	49
3.5	Conclusion	49
4	Quasi-réversibilité et nudging	51
4.1	Introduction sur la quasi-réversibilité	51
4.2	Présentation de la méthode	52
4.2.1	Heuristique	52
4.2.2	Justifications théoriques	52
4.3	Application à l'équation de la chaleur	54
4.3.1	Résultats théoriques	54
4.3.2	Expériences numériques	55
4.4	Variantes sur la quasi-réversibilité	58
4.4.1	Q.R. d'ordre 2 en temps	58
4.4.2	Q.R. d'ordre supérieur en temps	59
4.5	Autres systèmes différentiels testés	60
4.6	Conclusions sur la quasi-réversibilité	62
4.7	Introduction sur le nudging	62
4.8	Nudging direct	63
4.8.1	Principe de la méthode	63
4.8.2	Justification théorique dans un cadre linéaire	64
4.8.3	Remarques sur la méthode	64
4.9	Nudging rétrograde	65
4.10	Application au système de Lorenz	66
4.10.1	Présentation des équations du système de Lorenz	66
4.10.2	Nudging rétrograde appliqué au système de Lorenz	67
4.10.3	Résultats numériques du nudging rétrograde	67
4.11	Application à un modèle QG barotrope	69
4.11.1	Équations du modèle	69
4.11.2	Utilisation du nudging pour faire des prévisions	74
4.11.3	Reconstitution numérique de l'état initial	82
4.12	Conclusions sur le nudging	87
5	Dualité	89
5.1	Modèle physique	89
5.1.1	Quasi-géostrophie	89
5.1.2	Équations du modèle	90
5.1.3	Conditions aux bords	91
5.2	Algorithme 4D-VAR	92
5.2.1	Fonction coût	93
5.2.2	Modèle adjoint	93

5.2.3	Gradient	94
5.2.4	Erreur modèle	95
5.3	Algorithme dual	95
5.3.1	Généralités	95
5.3.2	Description théorique	96
5.3.3	Algorithme dual pour le modèle quasi-géostrophique	101
6	Résultats numériques	105
6.1	Paramètres numériques du modèle	105
6.1.1	Valeurs des coefficients du modèle	105
6.1.2	Paramètres des minimisations	105
6.2	Solution exacte, observations bruitées	106
6.2.1	Spin-up	107
6.2.2	Trajectoire de référence	107
6.2.3	Bruitage des observations	107
6.3	Convergence de la méthode duale	109
6.3.1	Convergence de la méthode duale	109
6.3.2	Comparaison avec la méthode primale sur un problème de faible dimension	112
6.4	Reconstitution de l'état initial et prévisions	114
6.4.1	Méthode primale	114
6.4.2	Méthode duale	116
6.4.3	Comparaison	116
6.5	Sensibilité à l'erreur modèle	119
6.5.1	Méthode primale	119
6.5.2	Méthode duale	121
6.5.3	Comparaison	122
6.6	Sensibilité aux observations	123
6.6.1	Méthode primale	123
6.6.2	Méthode duale	125
6.6.3	Comparaison	126
6.7	Détection de l'erreur modèle	128
6.7.1	Condition initiale exacte et observations exactes d'un modèle biaisé	128
6.7.2	Solutions identifiées par les deux méthodes	128
6.7.3	Comparaison des deux méthodes	130
6.8	Conclusions et remarques sur les deux méthodes	131
7	Conclusion	133

Chapitre 1

Introduction

La recherche scientifique repose sur deux sources d'information et d'étude différentes, les observations d'une part, et les modèles de l'autre. Au cours de l'histoire des Sciences, les observations ont d'abord été qualitatives puis se sont raffinées pour devenir quantitatives, alors que parallèlement, les modèles descriptifs se sont renforcés de l'apport des mathématiques, en particulier de l'Analyse Numérique.

Au cours des dernières décennies, les progrès dans l'étude de l'enveloppe fluide de la Terre, océans et atmosphère, ont été rendus possible par la conjonction systématique des observations et des apports de la théorie. La motivation et l'enjeu principal de ces deux sources de connaissances étaient et sont toujours le besoin immédiat de prévisions météorologiques numériquement calculables. Très rapidement, dans la recherche de méthodes adéquates, s'est imposée l'idée selon laquelle les modèles dynamiques utilisés devaient rendre compte de la propagation des informations dans le temps et dans l'espace. Cette notion fondamentale de propagation compense la disparité et l'imprécision des observations et permet de construire une image consistante, quadridimensionnelle, de l'atmosphère, ou plus généralement d'un fluide dont la taille rend compliquée voire impossible la description complète. L'un des enjeux de ce type d'étude est donc de confronter une nécessité d'information (météorologie ...), et la complexité des données : bruit, distribution très irrégulière des mesures dans le temps et dans l'espace. La combinaison de modèles numériques rendant compte des lois fondamentales de la Physique et des observations disponibles s'appelle *assimilation de données*, et donne lieu à des recherches actuellement en plein essor.

Les enjeux d'une telle connaissance prévisionnelle des océans et de l'atmosphère ne sont pas que purement scientifiques. Il y a en effet des enjeux environnementaux (traçabilité du dioxyde de carbone dans l'atmosphère, réchauffement de la planète, ...) ainsi qu'un aspect *grand public* : prévisions météorologiques accessibles à tous. Les prévisions ont également un rôle sécuritaire de plus en plus important, avec par exemple la mise en garde des populations contre d'éventuels événements climatiques hors normes (tempêtes, sécheresse, inondations, ...) et l'utilité de prévisions

océanographiques et météorologiques fiables à moyen terme pour le transport public (aviation civile, transport maritime, ...).

Conscients de l'importance de toutes ces applications de l'assimilation de données, de nombreux centres de recherche participent depuis plusieurs années à de vastes projets mondiaux ayant pour but non seulement la prévision opérationnelle, mais aussi l'étude de l'impact humain sur l'évolution du système océans-atmosphère et une meilleure connaissance des nombreux phénomènes climatiques régissant ce système. Cela est rendu possible par une amélioration, qualitative et quantitative, des sources d'information, notamment grâce aux nouvelles générations de satellites d'observation (European Remote Sensing ERS 2, Jason, Envisat), et aussi par une incessante augmentation des moyens de calcul. Malgré tout, de nombreuses difficultés demeurent (résolution des modèles encore mal adaptée à certains phénomènes turbulents, simplification des équations du modèles, paramètres physiques mal estimés, ...).

Il existe à ce jour deux grandes classes de méthodes d'assimilation de données qui correspondent à deux approches bien distinctes. Tout d'abord, l'assimilation *séquentielle*, qui repose sur des considérations statistiques et qui procède par corrections successives de la prévision du modèle au fur et à mesure que des observations sont disponibles ; et l'assimilation *variationnelle*, qui consiste à ajuster au mieux une solution du modèle à toutes les observations disponibles tout au long de la période d'assimilation.

Quelle que soit la méthode choisie, les problèmes sont globalement les mêmes, essentiellement celui de la taille du système, qui comprend généralement plus de 10^7 degrés de liberté, et qui pose certains soucis de temps de calcul et d'espace mémoire. La gestion des différentes erreurs commises dans le modèle et les observations est également délicate.

Les objectifs actuels en recherche sont donc les suivants : réduire la taille des systèmes étudiés sans perdre d'information, améliorer la connaissance des paramètres physiques et la statistique des erreurs de mesure, prendre en compte l'erreur modèle sans trop pénaliser la résolution numérique ...

Dans un premier temps, nous verrons dans le chapitre 2 quelques notions et définitions générales sur l'assimilation de données, les différentes méthodes actuelles, leurs intérêts et inconvénients ainsi que les projets opérationnels actuellement en cours d'élaboration à l'échelle mondiale.

Puis nous étudierons dans le chapitre 3 différents algorithmes de minimisation en vue d'une résolution effective d'un problème variationnel. Nous étudierons en détail l'algorithme L-BFGS (Limited memory Broyden-Fletcher-Goldfarb-Shanno), particulièrement adapté à des problèmes de grande taille et nous testerons ses différentes variantes sur l'équation de Burgers.

Le chapitre 4 est consacré à l'étude de deux méthodes servant à résoudre le problème inverse de l'identification de l'état initial. Il s'agit respectivement du nud-

ging (direct ou rétrograde), et de la quasi-réversibilité et certaines de ses variantes.

Le chapitre 5 introduit la notion de dualité dans l'assimilation de données en vue d'incorporer au modèle un terme d'erreur qui soit numériquement gérable. Nous donnons tout d'abord les équations du modèle physique utilisé pour les expériences, avant de comparer l'algorithme variationnel quadri-dimensionnel 4D-VAR et l'algorithme dual sur ce modèle.

Différentes expériences numériques ont été menées sur un modèle quasi-géostrophique barocline afin de comparer ces deux méthodes, dans plusieurs cas de figures : reconstitution de l'état initial, prévisions, sensibilité à l'erreur modèle, aux observations . . . Les différents résultats seront présentés dans le chapitre 6, qui précèdera une conclusion générale concernant ces différentes techniques liées de près ou de loin à l'assimilation de données.

Chapitre 2

L'assimilation de données

Nous allons introduire la notion d'assimilation de données avant d'exhiber une liste non exhaustive des différentes techniques qui peuvent être classées sous cette appellation. Nous verrons ensuite quelques utilisations de l'assimilation de données à des fins opérationnelles avant de pointer les principales difficultés actuelles et les enjeux qui font que l'assimilation de données est depuis quelques années en constant développement.

2.1 Qu'est-ce que l'assimilation de données ?

En océanographie et météorologie, les prévisions reposent largement sur une bonne estimation de l'état du système à un instant donné. Malheureusement, la technologie actuelle ne permet pas encore d'obtenir des informations fiables sur l'état du système, ne serait-ce que partiellement. Par exemple, il est évident qu'il n'est pas possible de mesurer à tout instant et partout la température qu'il fait en France. Or on souhaite connaître partout et au moins quelques jours à l'avance le temps qu'il fera. Les météorologues et océanographes ont pour ce faire un outil mathématique à leur disposition, l'assimilation de données. Celle-ci consiste à combiner de façon optimale les observations, récupérées par différents moyens (sondes, satellites), et la connaissance que nous avons du système qui régit les océans et l'atmosphère. Et ceci dans un but de pouvoir obtenir des prévisions fiables à plus ou moins court terme de leur évolution.

Pourquoi combiner ces deux sources d'informations ? D'une part les informations concernant les équations physiques des modèles qui gouvernent les océans et l'atmosphère sont incomplètes et imparfaites. En effet, les équations actuellement utilisées par les océanographes et les météorologues ne sont que des approximations plus ou moins réalistes des phénomènes physiques qui ont vraiment lieu, reposant sur des simplifications mathématiques et physiques et sur des résolutions numériquement

inexactes. D'un autre côté, pourquoi ne pas utiliser simplement toutes les observations dont nous disposons? En effet, elles sont nombreuses et de différents types : vitesses et trajectoires des vents mesurées par des stations au sol, des radiosondes, des ballons, des satellites...pour l'atmosphère; température, salinité, pression, vitesse des courants, dénivellation de la surface libre mesurées par des flotteurs fixes ou dérivants, bateaux, satellites, avions...pour les océans. Malheureusement, nombre de ces observations sont biaisées par des erreurs de mesure, plus ou moins importantes. Il est certain que les satellites de la génération actuelle (JASON, ENVISAT) sont nettement plus précis que les précédents (TOPEX/POSEIDON, European Remote Sensing ERS 1, ERS 2), mais il reste une incertitude de quelques pourcents. De plus, il reste des zones non (ou peu) couvertes par toutes ces mesures, et l'on souhaite tout de même pouvoir y faire des prévisions. Il est alors important de combiner ces deux sources d'informations et pouvoir utiliser la partie la plus fiable des observations et la propager dans le temps grâce aux modèles physiques.

L'assimilation de données en météorologie, qui ne consistait qu'à utiliser les observations des jours précédents pour essayer de mieux prévoir le temps qu'il ferait le lendemain, s'est considérablement développée et sert actuellement à plusieurs choses bien différentes. Tout d'abord, elle a toujours pour but de fournir de bonnes prévisions (à moyen ou long terme même comme en océanographie) de l'état du système étudié. Cela a un intérêt pour nombre de personnes : prévisions météo à quelques jours pour le grand public, prévisions météo à plus long terme sur une zone très particulière pour les organisateurs de grandes manifestations, état de l'océan pour les marins pêcheurs mais aussi pour l'armée, trajectoire des tempêtes pour les compagnies aériennes... Tout le monde utilise à divers degrés d'importance les prévisions météo et océanographiques. Mais l'assimilation de données ne s'arrête plus là. Elle a aussi pour but de mieux estimer certains paramètres des modèles physiques qui sont à l'heure actuelle encore inconnus, ou seulement partiellement identifiés. Cela permettra dans un futur proche d'améliorer considérablement les modèles physiques utilisés actuellement pour ces prévisions. La réassimilation de données passées (en connaissant plus ou moins précisément comment le système a évolué juste après) permet également de mieux comprendre certains phénomènes physiques qui jusqu'à présent étaient mal connus.

2.2 Différentes approches

À ce jour, il existe un grand nombre de méthodes dites d'assimilation de données, qui consistent toutes plus ou moins à fournir une estimation de la condition initiale du système, connaissant des estimations de ce même système à d'autres instants : elles consistent toutes à résoudre un problème dit *problème inverse*. Il existe actuellement deux grandes classes de méthodes d'assimilation de données, l'une reposant sur

des approches statistiques, l'assimilation *séquentielle* de données, et l'autre reposant sur la théorie du contrôle optimal [29], l'assimilation *variationnelle* de données. Dans un cadre purement linéaire, les techniques séquentielles et variationnelles sont très souvent équivalentes, mais leur extension à des problèmes non linéaires les différencie alors. Nous allons détailler dans ce chapitre plusieurs techniques séquentielles et variationnelles avant de nous limiter essentiellement dans les chapitres suivants à des techniques variationnelles.

Par la suite, nous allons considérer un système physique dont l'évolution est régie par un jeu d'équations différentielles qui peut s'écrire sous la forme :

$$\begin{cases} \frac{dx}{dt} = F(x), \\ x(0) = x_0, \end{cases} \quad (2.1)$$

où x , ne dépendant a priori que du temps, est le vecteur d'état. Nous noterons n sa dimension. $x_0 \in \mathbb{R}^n$ est la condition initiale de ce système différentiel. Dans la suite, nous noterons $y(t)$ les observations du système disponibles à l'instant t , et p la dimension du vecteur d'observations. À titre indicatif, pour un modèle (océanique ou météorologique) réaliste, n est de l'ordre de 10^7 et p de l'ordre de $10^5 - 10^6$. Soit H l'opérateur d'observation, a priori non linéaire et dépendant du temps, qui permet de relier le vecteur d'état x et les observations y . À partir d'un état du système $x \in \mathbb{R}^n$, il permet de générer un vecteur $Hx \in \mathbb{R}^p$ pouvant être assimilé (et donc comparé) à une observation.

2.2.1 Assimilation séquentielle

L'assimilation séquentielle de données repose sur des études statistiques des états des systèmes afin de trouver celui qui, statistiquement, correspond le mieux aux observations. Les méthodes séquentielles reposent essentiellement soit sur l'interpolation optimale, soit sur le filtre de Kalman.

Interpolation optimale

Le principe de l'interpolation optimale est de chercher une combinaison linéaire optimale entre les observations et les états du système aux mêmes instants. L'estimateur qui réalise le minimum de la variance de l'erreur d'estimation est alors appelé *BLUE*, Best Linear Unbiased Estimator ([42], [31], [12], [36]). Si on note x_b (background ou ébauche) une estimation de l'état du système x avant assimilation de données, et x_a (analysis ou état analysé) le BLUE, le but est de chercher le meilleur terme correctif à x_b en fonction du vecteur d'innovation, le vecteur $y - y_b$, qui représente l'écart entre les observations et l'état correspondant du système.

Il y a deux façons de trouver cette meilleure estimation. La première repose essentiellement sur des considérations statistiques et l'autre plutôt sur la minimisation d'une fonctionnelle, mais ces deux approches sont en fait très comparables. Étudions tout d'abord l'approche statistique.

Étant données deux variables aléatoires x et y d'espérances mathématiques respectives x_b et y_b , la meilleure estimation linéaire de x à partir de y (qui minimise la variance de l'erreur d'estimation) est :

$$x_a - x_b = E((x - x_b)(y - y_b)^T) \cdot E((y - y_b)(y - y_b)^T)^{-1} \cdot (y - y_b). \quad (2.2)$$

Il apparaît alors que $x_a - x_b$ est la projection orthogonale (au sens de la covariance statistique) de x sur l'espace vectoriel engendré par y , et par conséquent, $x - x_a$ et $y - y_b$ sont orthogonaux et donc décorrélés.

Nous allons désormais considérer que y_b est relié à l'ébauche x_b par une relation linéaire de la forme $y_b = Hx_b$, H pouvant être considéré comme un opérateur d'observation servant à relier les états du système x aux observations y . En notant $B = E((x - x_b)^T(x - x_b))$ la matrice de covariance de l'erreur d'ébauche, et $R = E((y - y_b)^T(y - y_b))$ la matrice de covariance d'erreur d'observation, et en supposant que les erreurs d'ébauche et d'observation sont décorrélées, on peut réécrire (2.2) sous la forme suivante :

$$x_a = x_b + BH^T(HBH^T + R)^{-1} \cdot (y - Hx_b). \quad (2.3)$$

L'équation (2.3) donne le meilleur estimateur possible (le *BLUE*) de x connaissant le vecteur d'ébauche x_b , le vecteur d'innovation $y - y_b$ et les matrices de covariance des erreurs relatives à l'estimation de l'état réel B et aux mesures des observations R .

Si on s'intéresse au même problème, mais en essayant de quantifier l'écart du vecteur d'état x par rapport à x_b et celui de l'observation y par rapport à Hx , nous pouvons introduire la fonction coût suivante :

$$J(x) = \frac{1}{2}(x - x_b)^T B^{-1}(x - x_b) + \frac{1}{2}(y - Hx)^T R^{-1}(y - Hx). \quad (2.4)$$

La première partie de la fonctionnelle J mesure l'écart au sens des moindres carrés (via la matrice de covariance d'erreur d'ébauche) entre l'état du système x et l'ébauche x_b . Le second terme mesure de la même façon l'écart aux observations. Le minimum d'une telle fonction coût devrait ainsi être proche à la fois de l'ébauche et des observations.

En supposant l'opérateur d'observation H linéaire, la fonctionnelle J est strictement convexe et son minimum est alors atteint lorsque son gradient est nul :

$$\nabla J(x_a) = B^{-1}(x_a - x_b) - H^T R^{-1}(y - Hx_a) = 0 \quad (2.5)$$

ce qui donne

$$\begin{aligned} x_a &= (B^{-1} + H^T R^{-1} H)^{-1} (B^{-1} x_b + H^T R^{-1} y) \\ &= x_b + (B^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1} (y - H x_b). \end{aligned}$$

Un calcul rapide portant sur la matrice de gain $(B^{-1} + H^T R^{-1} H)^{-1} H^T R^{-1}$ montre que cette formule est équivalente à

$$x_a = x_b + B H^T (H B H^T + R)^{-1} (y - H x_b), \quad (2.6)$$

ce qui est exactement la même équation que (2.3).

Cette méthode est relativement *rapide*, puisqu'il suffit de multiplier le vecteur d'innovation par une matrice (matrice de gain) pour obtenir le terme correctif à apporter à l'ébauche pour trouver la meilleure estimation linéaire de l'état réel du système. Ceci dit, elle repose essentiellement sur la connaissance des deux matrices de covariance d'erreur, ce qui est loin d'être le cas, ainsi que sur la linéarité de l'opérateur d'observation H , ce qui n'est pas toujours vrai. Enfin, la taille des matrices n'est pas sans poser quelques difficultés numériques.

Filtre de Kalman

Nous allons désormais nous pencher sur la technique du filtrage de Kalman ([22], [16], [17], [18], [20], [15], [14], [8], [52]). Cette théorie fournit à chaque nouvelle observation une nouvelle estimation de la variable d'état. Le filtrage de Kalman utilise les observations dans l'ordre où elles sont disponibles, d'où l'appellation séquentielle. Par conséquent, l'estimation de l'état du système ne dépendra pas des observations futures mais uniquement des observations passées et présentes.

Considérons une discrétisation en temps de notre problème :

$$\begin{cases} x_0 \text{ donné,} \\ x_{n+1} = M_n x_n + \varepsilon_n, \quad n \geq 0, \end{cases} \quad (2.7)$$

où M_n est une approximation linéaire du modèle à l'instant t_n correspondant au $n^{\text{ième}}$ pas de temps et ε_n est l'incertitude sur le modèle à cet instant.

On doit également discrétiser en temps les observations :

$$y_n = H_n x_n + o_n, \quad (2.8)$$

où H_n est une approximation linéaire (si H ne l'est pas) de l'opérateur d'observation à l'instant t_n et o_n représente l'erreur d'observation à ce même instant.

Notons, à chaque instant t_n , Q_n et R_n les matrices de covariance des erreurs de modèle et d'observation respectivement. Nous supposons dans la suite que ces deux

erreurs sont d'espérance nulle et décorréées (l'une par rapport à l'autre et entre elles à des instants différents).

L'ébauche de l'état du système à l'instant t_n n'est autre que l'état obtenu en faisant évoluer l'analyse réalisée à l'instant d'avant sur un pas de temps :

$$x_n^f = M_{n-1}x_{n-1}^a \quad (2.9)$$

avec x_{n-1}^a l'état analysé à l'instant t_{n-1} .

Si on note P_n^f la matrice de covariance de l'erreur d'ébauche $x_n^f - x_n$ et P_n^a la matrice de covariance de l'erreur d'analyse $x_n^a - x_n$, alors

$$\begin{aligned} P_{n+1}^f &= E \left((x_{n+1}^f - x_{n+1})^T (x_{n+1}^f - x_{n+1}) \right) \\ &= E \left((M_n x_n^a - M_n x_n - \varepsilon_n)^T (M_n x_n^a - M_n x_n - \varepsilon_n) \right), \end{aligned}$$

donc

$$P_{n+1}^f = M_n P_n^a M_n^T + Q_n. \quad (2.10)$$

Cette phase est appelée la phase de prédiction : on a déduit une estimation de l'état à l'instant t_{n+1} de la prévision de l'instant t_n , ainsi que la matrice de covariance de l'erreur d'ébauche en fonction de la matrice de covariance de l'erreur d'analyse de l'instant précédent.

À partir de cette ébauche et du vecteur d'innovation $y_{n+1} - H_{n+1}x_{n+1}^f$, il faut construire l'état analysé x_{n+1}^a , de façon analogue à la méthode de l'interpolation optimale :

$$x_{n+1}^a = x_{n+1}^f + K_{n+1}(y_{n+1} - H_{n+1}x_{n+1}^f) \quad (2.11)$$

où K_{n+1} est la matrice de gain, dite de Kalman. Cette matrice est définie par :

$$K_{n+1} = P_{n+1}^f H_{n+1}^T \left(H_{n+1} P_{n+1}^f H_{n+1}^T + R_{n+1} \right)^{-1}. \quad (2.12)$$

Les équations (2.11) et (2.12) ne sont pas sans rappeler l'équation (2.6) qui donnait l'analyse en fonction de l'ébauche dans le cadre de l'interpolation optimale, sauf qu'ici la dépendance du temps intervient dans chaque matrice et chaque vecteur.

Il est alors aisé de calculer la nouvelle matrice de covariance d'erreur d'analyse :

$$P_{n+1}^a = P_{n+1}^f - K_{n+1} H_{n+1} P_{n+1}^f. \quad (2.13)$$

Cette phase est appelée la phase de correction : on a corrigé l'estimation fournie par l'étape de prédiction à l'aide du vecteur d'innovation et d'une matrice de gain.

Il faut noter que la matrice de covariance d'erreur d'analyse P_{n+1}^a est indépendante des valeurs des observations, contrairement à l'état analysé x_{n+1}^a .

Le gros avantage de la méthode du filtre de Kalman est de fournir à chaque itération une estimation des matrices de covariance d'erreur d'ébauche et d'analyse.

Il faut toutefois initialiser correctement ces matrices à l'instant t_0 , et avoir une estimation des matrices de covariance d'erreur modèle et d'erreur d'observation. Cependant, en météorologie, l'initialisation n'est pas une difficulté majeure car les observations s'accumulent dans le temps, et en pratique, la mise en route de l'algorithme de filtrage est assez vite oubliée.

Plusieurs inconvénients majeurs sont cependant connus. Tout d'abord, dans le cas d'un modèle non linéaire, le filtre de Kalman ainsi défini n'est pas optimal (il ne fournit pas la solution de variance minimale). Pour pallier à ce défaut qui pose problème dès qu'on utilise le filtre de Kalman sur un problème même faiblement non linéaire, a été introduit le filtre de Kalman étendu qui travaille à chaque instant sur des versions linéarisées du modèle (système linéaire tangent) et de l'opérateur d'observation (opérateur linéaire tangent d'observation). Ainsi, dans le cas d'un modèle linéaire, il est optimal, et dans le cas d'un modèle non linéaire, même s'il n'est pas forcément optimal, il donne de bonnes analyses si les non linéarités ne sont pas trop importantes.

Le second problème qui se pose est le coût de la mise en œuvre numérique du filtre de Kalman. En effet, si la dimension du problème est de l'ordre de 10^7 , et la dimension des observations de l'ordre de 10^5 , les matrices P_n^f , P_n^a , M_n et Q_n sont des matrices carrées de dimension $10^7 \times 10^7$, la matrice R_n est carrée de dimension $10^5 \times 10^5$ et les matrices H_n et K_n sont rectangulaires de dimension $10^5 \times 10^7$. Le stockage d'une matrice de cet ordre de grandeur pose problème, mais il n'est heureusement pas utile, car seul le résultat de la multiplication d'une telle matrice par un vecteur est nécessaire. Pour des raisons évidentes de coût de calcul, il est nécessaire de réduire la dimension du filtre si on veut l'appliquer à des problèmes d'océanographie ou météorologie.

Filtre SEEK

Le filtre SEEK (Singular Evolutive Extended Kalman) a été récemment introduit ([41]) pour pallier à ces difficultés. Il prend en compte à la fois les versions linéarisées des opérateurs pour une application à un système non linéaire (Extended) et une réduction d'ordre (Singular Evolutive), généralement à l'aide d'une analyse EOF (Empirical Orthogonal Functions). Au lieu de travailler sur l'espace d'état de dimension trop élevée, le filtre SEEK travaille avec des matrices de dimensions plus petites, représentées par des vecteurs décrivant les modes dominants du système. Généralement on ne conserve que quelques dizaines de modes propres, ce qui réduit considérablement le rang des matrices, et permet d'envisager une mise en œuvre numérique.

Beaucoup d'autres évolutions du filtre de Kalman existent (Kalman ensemble, Kalman adaptatif, . . .), chacune apportant quelques simplifications au filtre de Kal-

man étendu de façon à le rendre numériquement programmable pour des problèmes de grande dimension, généralement en réduisant la taille des matrices de covariance.

2.2.2 Assimilation variationnelle

Nous allons nous intéresser maintenant aux méthodes variationnelles ([29], [27], [25], [48], [49], [46], [37], [45], [38], [35]). Celles-ci, par opposition aux méthodes séquentielles qui ne traitaient les observations qu'au fur et à mesure qu'elles étaient disponibles sans jamais utiliser une observation future dans une analyse, traitent le problème globalement, sous forme de minimisation d'une fonction coût contenant à la fois les informations relatives au modèle et les observations. Cette fonctionnelle mesure généralement les écarts au sens des moindres carrés entre les états estimés du système et les données. Ces méthodes nécessitent également une bonne connaissance statistique des diverses erreurs afin d'affecter un certain poids à chaque terme de la fonctionnelle à minimiser.

Nous allons maintenant étudier quelques-unes des méthodes variationnelles les plus connues et les plus utilisées : le 3D-VAR (variationnel tri-dimensionnel), le 4D-VAR (quadri-dimensionnel), le 4D-VAR incrémental et le 4D-PSAS (Physical Space Analysis System).

3D-VAR

Nous notons encore x_b l'ébauche de la condition initiale, y le vecteur des observations du système, H l'opérateur d'observations permettant de relier un état x du système aux observations y . La méthode du 3D-VAR (méthode variationnelle à 3 dimensions) consiste à minimiser une fonction coût qui mesure à la fois l'erreur sur la condition initiale et l'erreur sur les observations :

$$J(x) = \frac{1}{2}(x - x_b)^T B^{-1}(x - x_b) + \frac{1}{2}(y - H(x))^T R^{-1}(y - H(x)), \quad (2.14)$$

où B et R sont les matrices de covariance d'erreur sur respectivement la condition initiale et les observations. La minimisation s'effectue en général avec un algorithme de type BFGS (voir le chapitre suivant) qui ne nécessite que la connaissance du gradient de la fonction coût pour la minimiser. En supposant que H est linéaire, celui-ci est évidemment :

$$\nabla J(x) = B^{-1}(x - x_b) - H^T R^{-1}(y - H(x)). \quad (2.15)$$

Lorsque des algorithmes de type Newton sont utilisés (uniquement lorsque la dimension du problème le permet), il faut également connaître le hessien de la fonction coût

$$\nabla^2 J(x) = B^{-1} + H^T R^{-1} H, \quad (2.16)$$

indépendant de x (toujours dans l'hypothèse où H est linéaire).

Le point de départ de la minimisation est souvent une ébauche de la condition initiale, résultat d'une prévision passée. Lorsque la minimisation aboutit (le gradient de la fonctionnelle a suffisamment diminué), le minimum trouvé est une bonne approximation de l'état initial réel puisqu'il est proche de l'ébauche et des observations du système.

3D-VAR incrémental La version incrémentale du 3D-VAR consiste à considérer comme contrôle l'écart entre l'état du système et l'ébauche, et non plus l'état du système :

$$\delta x = x - x_b.$$

La fonction coût devient alors :

$$J(\delta x) = \frac{1}{2} \delta x^T B^{-1} \delta x + \frac{1}{2} (d - H(\delta x))^T R^{-1} (d - H(\delta x))$$

où d est le vecteur d'innovation $y - H(x_b)$.

La minimisation de cette fonctionnelle est semblable à celle du 3D-VAR. Le gros avantage de la réécriture du problème sous cette forme est la réduction possible de la dimension de l'espace de contrôle, et donc de la dimension des matrices de covariance, à l'aide d'hypothèses simplificatrices sur l'écart entre l'état du système et l'ébauche.

Ces méthodes tri-dimensionnelles relativement simples et efficaces étaient opérationnelles à Météo France et à l'ECMWF il y a encore quelques années ([10], [44], [2]). Elles ont été remplacées depuis par le 4D-VAR.

4D-VAR

Le 4D-VAR est une généralisation du 3D-VAR en ajoutant une dimension supplémentaire au problème de minimisation, le temps. En effet, on va considérer désormais que les observations sont distribuées en temps, et les traiter séparément à divers instants.

Considérons à nouveau un système régi par une équation différentielle du type :

$$\begin{cases} \frac{dx}{dt} = F(x), \\ x(0) = x_0, \end{cases} \quad (2.17)$$

où x_0 est mal connu.

Notons (t_i) , $0 \leq i \leq n$, les instants où des observations y_i sont disponibles. Notons H_i les opérateurs d'observation correspondants et R_i leur matrice de covariance

d'erreur. On peut alors définir une fonction coût ainsi :

$$J(x_0) = \frac{1}{2}(x_0 - x_b)^T B^{-1}(x_0 - x_b) + \frac{1}{2} \sum_{i=0}^n (y_i - H_i(x(t_i)))^T R_i^{-1} (y_i - H_i(x(t_i))) \quad (2.18)$$

où $x(t_i)$ est l'état du système obtenu en faisant évoluer les équations entre l'instant initial, avec x_0 comme condition initiale, et t_i . Les équations du modèle interviennent donc de façon implicite dans la définition de cette fonctionnelle qui ne dépend que de la condition initiale x_0 . Ce sont néanmoins des contraintes fortes.

La minimisation de cette fonctionnelle requiert la connaissance de son gradient. Celui-ci n'est plus évident à trouver, comme dans le cas du 3D-VAR. En effet, comment dépendent les termes $x(t_i)$ de la condition initiale x_0 ? Il faudrait, par une technique de différences finies, résoudre le système différentiel (2.17) au moins autant de fois que la dimension du vecteur x_0 . Or, celle-ci dépasse souvent 10^6 , et il n'est pas envisageable de résoudre des millions de fois un système déjà coûteux à résoudre une fois. Pour parer à cet inconvénient majeur, Le Dimet et al. [25] ont introduit en 1986 la méthode de l'adjoint qui permet, en une seule résolution d'un système équivalent au système (2.17), de trouver le gradient de la fonctionnelle.

Adjoint au premier ordre Considérons une petite perturbation $\delta x_0 = \alpha h$ de la condition initiale x_0 , où α est le paramètre que nous ferons tendre vers 0 et h la direction de la perturbation. Soit \tilde{x} la solution du système

$$\begin{cases} \frac{d\tilde{x}}{dt} = F(\tilde{x}), \\ \tilde{x}(0) = x_0 + \delta x_0. \end{cases} \quad (2.19)$$

Si on fait la différence entre la solution \tilde{x} de (2.19) et la solution x de (2.17), puis en divisant cela par α , $\delta x = \frac{\tilde{x} - x}{\alpha}$ vérifie le système différentiel suivant :

$$\begin{cases} \frac{d \delta x}{dt} = [F'(x)] \delta x, \\ \delta x(0) = h. \end{cases} \quad (2.20)$$

δx est alors la dérivée de x le long de la direction h . La dérivée de J par rapport à x_0 dans la direction h est alors :

$$\nabla J(x_0).h = (x_0 - x_b)^T B^{-1} h - \sum_{i=0}^n (y_i - H_i(x(t_i)))^T R_i^{-1} H'_i(x(t_i)) \delta x(t_i). \quad (2.21)$$

En définissant l'état adjoint (voir [47] pour la construction des équations adjointes à partir des équations du modèle) :

$$\begin{cases} -\frac{dp}{dt} = [F'(x)]^T p - \sum_{i=0}^n H'_i(x(t_i))^T R_i^{-1} (y_i - H_i(x(t_i))) \delta_{t_i}(t), \\ p(T) = 0 \end{cases} \quad (2.22)$$

avec une condition finale nulle, il vient :

$$\begin{aligned} \nabla J(x_0).h &= (x_0 - x_b)^T B^{-1} h - \int_0^T \sum_{i=0}^n H'_i(x(t_i))^T R_i^{-1} (y_i - H_i(x(t_i))) \delta_{t_i}(t). \delta x(t) dt \\ &= (x_0 - x_b)^T B^{-1} h - \int_0^T \left([F'(x)]^T p(t) + \frac{dp}{dt}(t) \right) . \delta x(t) dt \\ &= (x_0 - x_b)^T B^{-1} h - \int_0^T \left(p(t). \frac{d \delta x}{dt}(t) + \frac{dp}{dt}(t). \delta x(t) \right) dt \\ &= (x_0 - x_b)^T B^{-1} h - [p(t). \delta x(t)]_0^T \\ &= (x_0 - x_b)^T B^{-1} h - p(T). \delta x(T) + p(0). \delta x(0) \\ &= (x_0 - x_b)^T B^{-1} h + p(0). h \end{aligned}$$

On en déduit donc le gradient de la fonction coût :

$$\nabla J(x_0) = B^{-1}(x_0 - x_b) + p(0), \quad (2.23)$$

p étant l'état adjoint, solution de (2.22).

Si on utilise un algorithme de minimisation qui ne nécessite que la connaissance de la fonction coût et de son gradient, à chaque itération, il ne faut que deux intégrations d'un système différentiel : le système direct (2.17) afin de calculer J , et le système adjoint (2.22) afin de calculer ∇J .

Adjoint au second ordre Certains algorithmes de descente nécessitent aussi la connaissance du hessien de la fonction coût. Nous allons voir rapidement la méthode de l'adjoint au second ordre qui donne le produit du hessien par un vecteur donné en une seule intégration supplémentaire d'une équation différentielle, là où une méthode classique de différences finies en aurait nécessité au moins autant que la dimension des variables.

Considérons une perturbation δx_0 de la condition initiale x_0 . La perturbation résultante de x est donnée au premier ordre par le modèle linéaire tangent

$$\begin{cases} \frac{d \delta x}{dt} = F'(x) \delta x, \\ \delta x(0) = \delta x_0, \end{cases} \quad (2.24)$$

et la perturbation correspondant δp de la variable adjointe est alors donnée par le modèle adjoint au second ordre :

$$\left\{ \begin{array}{l} -\frac{d \delta p}{dt} = [F'(x)]^T \delta p + [F''(x) \delta x] p \\ \quad + \sum_{i=0}^n \left\{ [H'_i(x(t_i))]^T R_i^{-1} H'_i(x(t_i)) \right. \\ \quad \quad - \frac{1}{2} [H''_i(x(t_i))]^T R_i^{-1} (y_i - H_i(x(t_i))) \\ \quad \quad \quad \left. - \frac{1}{2} (y_i - H_i(x(t_i)))^T R_i^{-1} [H''_i(x(t_i))] \right\} \delta x \delta_{t_i}(t), \\ \delta p(T) = 0. \end{array} \right. \quad (2.25)$$

Si on note $p_{x_0+\delta x_0}$ la solution du modèle adjoint au premier ordre après la perturbation δx_0 de la condition initiale x_0 du modèle direct, par définition, on a $p_{x_0+\delta x_0}(0) = p(0) + \delta p(0)$, et donc

$$\begin{aligned} \nabla^2 J(x_0) \cdot \delta x_0 &= \nabla J(x_0 + \delta x_0) - \nabla J(x_0) + \mathcal{O}(\|\delta x_0\|^2) \\ &= (B^{-1}(x_0 + \delta x_0 - x_b) + p_{x_0+\delta x_0}(0)) \\ &\quad - (B^{-1}(x_0 - x_b) + p(0)) \end{aligned}$$

et donc

$$\nabla^2 J(x_0) \cdot \delta x_0 = B^{-1} \delta x_0 + \delta p(0). \quad (2.26)$$

4D-VAR incrémental

Là encore, comme pour le 3D-VAR, une version incrémentale permet de réduire la dimension du vecteur de contrôle, en ne travaillant plus sur la condition initiale x_0 mais sur l'incrément $\delta x_0 = x_0 - x_b$ ([11]). En effet, le 4D-VAR tel que nous venons de le voir a de nombreux inconvénients : la dimension est très élevée (de l'ordre de 10^6 au moins), l'écriture du code adjoint (et éventuellement de l'adjoint au second ordre, si nécessaire) est fastidieuse, mais nécessaire dans tous les algorithmes quadri-dimensionnels que nous étudions ici, y compris dans le 4D-PSAS. Les matrices de covariance d'erreur ne sont à aucun moment fournies par l'algorithme, mais ce n'est pas le but de cet algorithme variationnel. Enfin, la fonction coût est non linéaire et le coût de chaque itération (résolution des équations directes et adjointes) rend la minimisation difficile.

La fonctionnelle incrémentale est la suivante :

$$J(\delta x_0) = \frac{1}{2} \delta x_0^T B^{-1} \delta x_0 + \frac{1}{2} \sum_{i=0}^n [d_i - H_i(\delta x(t_i))]^T R_i^{-1} [d_i - H_i(\delta x(t_i))] \quad (2.27)$$

où $d_i = y_i - H_i(x_b(t_i))$, différence entre l'observation y_i et la solution à l'instant t_i de l'équation directe (2.17) avec pour condition initiale x_b , est le vecteur d'innovation correspondant à la $i^{\text{ème}}$ observation, et $\delta x(t_i)$ est la solution à l'instant t_i du système linéaire tangent avec δx_0 comme condition initiale. Il faut également supposer que H_i est linéaire (si ce n'est pas le cas, il faut le remplacer par une approximation linéaire). Une fois la fonction coût incrémentale minimisée, il suffit d'ajouter la solution trouvée à l'ébauche pour reconstituer l'état initial x_0 .

En choisissant quelques modes de grande variabilité (quelques dizaines), il est possible de ramener le problème initial (comme traité par le 4D-VAR) de très grande taille à un problème de taille très raisonnable, résoluble par l'algorithme 4D-VAR incrémental. De plus, comme dans les autres algorithmes variationnels, il est toujours possible d'arrêter la minimisation effective de la fonctionnelle bien avant convergence, lorsque le temps de calcul atteint une éventuelle limite fixée. Ce sont les principaux avantages de cette méthode, et c'est pour cela que de nombreux organismes de météorologie l'ont adopté dans leur système de prévision ([37], [45], [35]).

4D-PSAS (Physical Space Analysis System)

À ce jour, le principal inconvénient du 4D-VAR (et de sa version incrémentale) est qu'il suppose le modèle parfait, et il ne peut à aucun moment prendre en compte une erreur modèle. En effet, cela reviendrait à rajouter un terme inconnu, dépendant du temps, au modèle direct (ainsi qu'au linéaire tangent et à l'adjoint). Il faudrait ensuite ajouter ces termes au vecteur de contrôle, et le problème de minimisation serait alors très délicat à résoudre, sa taille pouvant facilement dépasser 10^9 . En effet, même si le nombre d'itérations à effectuer dans l'algorithme de minimisation est indépendant de la taille du problème, le coût de chaque itération augmente.

L'algorithme 4D-PSAS, encore appelé algorithme dual, permet de prendre en compte de façon inhérente l'erreur modèle ([4], [5], [1], [9], [33], [32]). Nous allons expliquer sommairement son principe, car nous reviendrons dessus en détail dans le chapitre 6.

Rajoutons une erreur dans le modèle discrétisé :

$$x_{i+1} = M_i x_i + \varepsilon_i, \quad i \geq 0, \quad (2.28)$$

M_i désignant, comme dans le paragraphe sur le filtre de Kalman, une approximation linéaire du modèle à l'instant t_i . ε_i représente l'erreur modèle à cet instant. Notons o_i l'erreur d'observation :

$$y_i = H_i x_i + o_i, \quad (2.29)$$

avec H_i une approximation linéaire de l'opérateur d'observation à l'instant t_i .

Il faut définir une nouvelle fonction coût, intégrant les erreurs modèle :

$$\begin{aligned}
J(x_0, \varepsilon_0, \dots, \varepsilon_n) &= \frac{1}{2}(x_0 - x_b)^T B^{-1}(x_0 - x_b) \\
&+ \frac{1}{2} \sum_{i=0}^n (y_i - H_i(x(t_i)))^T R_i^{-1} (y_i - H_i(x(t_i))) \\
&+ \frac{1}{2} \sum_{i=0}^n \varepsilon_i^T Q_i^{-1} \varepsilon_i
\end{aligned} \tag{2.30}$$

où Q_i est la matrice de covariance de l'erreur modèle à l'instant t_i .

Comme nous l'avons précédemment vu, il est difficile de minimiser une telle fonction, simplement au vu de la taille et du nombre des vecteurs inconnus.

Plaçons nous dans une optique lagrangienne, où l'on chercherait à minimiser la fonctionnelle J précédente. En écriture incrémentale, on peut définir le lagrangien suivant :

$$\begin{aligned}
\mathcal{L}(\delta x_0, (\varepsilon_i)_{0 \leq i \leq n}, (\mathcal{Y}_i)_{0 \leq i \leq n}; (m_i)_{0 \leq i \leq n}) &= \frac{1}{2} \delta x_0^T B^{-1} \delta x_0 \\
&+ \frac{1}{2} \sum_{i=0}^n (d_i - \mathcal{Y}_i)^T R_i^{-1} (d_i - \mathcal{Y}_i) \\
&+ \frac{1}{2} \sum_{i=0}^n \varepsilon_i^T Q_i^{-1} \varepsilon_i \\
&+ \sum_{i=0}^n [\mathcal{Y}_i - H_i(\mathcal{M}_i \delta x_0 + \tilde{\varepsilon}_i)] . m_i
\end{aligned} \tag{2.31}$$

où $d_i = y_i - H_i(x_b(t_i))$ est le vecteur d'innovation à l'instant t_i et $\mathcal{Y}_i = H_i(\delta x(t_i))$ est le représentant de l'état du système à l'instant t_i dans l'espace des observations. Si on suppose que \mathcal{Y}_i est désormais une inconnue, il faut rajouter à la fonctionnelle la contrainte de modèle portant sur \mathcal{Y}_i : $\delta x(t_i) = \mathcal{M}_i \delta x_0 + \tilde{\varepsilon}_i$, en notant \mathcal{M}_i le modèle linéaire tangent permettant de passer de l'instant 0 à l'instant t_i , et $\tilde{\varepsilon}_i$ l'erreur correspondante. m_i désigne le multiplicateur de Lagrange associé à la $i^{\text{ème}}$ contrainte.

Le modèle apparaît désormais comme une contrainte faible et n'est plus une contrainte forte comme dans le 4D-VAR. On a la propriété suivante (voir [43] pour les problèmes de min-max et d'optimisation sous contrainte) :

$$\min_{\delta x_0, (\varepsilon_i)} J(x_0, (\varepsilon_i)_{0 \leq i \leq n}) = \min_{\delta x_0, (\varepsilon_i)} \max_{(m_i)} \mathcal{L}(\delta x_0, (\varepsilon_i)_{0 \leq i \leq n}, (\mathcal{Y}_i)_{0 \leq i \leq n}; (m_i)_{0 \leq i \leq n}) \tag{2.32}$$

Il ne reste plus qu'à changer l'écriture de (2.32) pour obtenir un problème de maximisation (de la fonctionnelle duale), équivalent au problème de minimisation de

la fonctionnelle primale (2.30) :

$$\begin{aligned}
& \min_{\delta x_0, (\varepsilon_i)} \max_{(m_i)} \mathcal{L}(\delta x_0, (\varepsilon_i)_{0 \leq i \leq n}, (\mathcal{Y}_i)_{0 \leq i \leq n}; (m_i)_{0 \leq i \leq n}) \\
&= \max_{(m_i)} \min_{\delta x_0, (\varepsilon_i)} \mathcal{L}(\delta x_0, (\varepsilon_i)_{0 \leq i \leq n}, (\mathcal{Y}_i)_{0 \leq i \leq n}; (m_i)_{0 \leq i \leq n}) \\
&= \max_{(m_i)} [-J_{\mathcal{D}}((m_i)_{0 \leq i \leq n})]
\end{aligned} \tag{2.33}$$

Il faut noter que la réécriture du problème consistant à minimiser la fonctionnelle J sous forme d'un problème de min-max ainsi que la réécriture duale du problème ne nécessite pas que le problème soit linéaire, toute cette théorie reste valable lorsque le problème est convexe.

Quitte à changer le signe du résultat, on se ramène à devoir minimiser la fonctionnelle duale suivante :

$$\begin{aligned}
J_{\mathcal{D}}((m_i)_{0 \leq i \leq n}) &= \frac{1}{2} \sum_{i=0}^n [\mathcal{M}_i^T H_i^T m_i]^T B_i [\mathcal{M}_i^T H_i^T m_i] \\
&+ \frac{1}{2} \sum_{i=0}^n m_i^T R_i m_i \\
&+ \frac{1}{2} \sum_{i=0}^n [H_i^T m_i]^T Q_i [H_i^T m_i] \\
&- \sum_{i=0}^n d_i \cdot m_i
\end{aligned} \tag{2.34}$$

Sous forme matricielle, cela peut s'écrire

$$J_{\mathcal{D}}(m) = \frac{1}{2} m^T (\mathcal{D} + R) m - d^T m \tag{2.35}$$

avec $\mathcal{D} = HMBM^T H^T + HQH^T$, appelée matrice des représentants car chacun de ses coefficients agit sur une et une seule observation. Dans un cadre linéaire, la fonctionnelle duale est quadratique et donc facile à minimiser, son gradient étant évident. La théorie classique de la dualité assure de plus que les deux fonctionnelles, primale (2.30) et duale (2.34), ont même minimum, et que ce minimum est atteint au même point.

L'autre point intéressant de cet algorithme est que le vecteur de contrôle dual (m) appartient à l'espace des observations qui est de dimension souvent 10 fois plus petite que l'espace des états pour la fonctionnelle primale. En effet, pour un modèle opérationnel, la dimension du vecteur d'état est compris entre 10^6 et 10^7 alors que les observations utilisables sont en général de l'ordre de 10^5 pour les périodes d'assimilation usuelles. La minimisation intervient donc sur un espace plus petit, il est donc

possible d'obtenir de meilleurs résultats pour un nombre comparable d'itérations dans l'algorithme de descente. En effet, chaque itération dans la minimisation a globalement le même coût que pour le 4D-VAR puisque le conditionnement du problème est sensiblement le même, mais pour le même coût, le 4D-PSAS prend en compte l'erreur modèle.

Enfin, l'algorithme dual utilise le modèle comme une contrainte faible, mais il suffit de faire tendre les erreurs modèle ε_i vers 0 pour retrouver à la limite l'algorithme primal (4D-VAR) à contrainte forte. Ce passage à la limite ne pose aucune difficulté puisque le 4D-PSAS utilise les matrices de covariance directes (et non leurs inverses comme dans le cas du 4D-VAR).

Le 4D-PSAS est donc une alternative intéressante au 4D-VAR puisqu'il conduit, en théorie, aux mêmes résultats, mais l'algorithme dual est le seul qui permet en pratique en prendre en compte l'erreur modèle. Pour ces raisons, cet algorithme est probablement celui qui devrait équiper prochainement les centres européens de prévision météo.

Remarques générales sur les méthodes séquentielles et variationnelles

Dans un cadre strictement linéaire, les méthodes séquentielles du type filtre de Kalman et les méthodes variationnelles sont deux approches distinctes pour résoudre le même problème, l'équation (2.2), c'est-à-dire trouver la meilleure estimation linéaire de l'état du système à partir des observations. Ces deux classes de méthodes ne sont bien sûr plus équivalentes lorsque le problème est non linéaire. De plus, pour que le filtre de Kalman soit optimal, les erreurs doivent être décorréélées en temps, alors que les méthodes variationnelles n'ont pas besoin de cette hypothèse.

2.3 Difficultés actuelles et objectifs

2.3.1 Difficultés et enjeux

Chacune des méthodes d'assimilation de données présentées dans ce chapitre a son lot de difficultés, plus ou moins faciles à gérer. Il ressort globalement deux principaux défauts communs à toutes ces techniques.

La première concerne la non linéarité. En effet, que ce soient les méthodes séquentielles ou les méthodes variationnelles, elles doivent toutes à un moment ou un autre utiliser les versions linéarisées des modèles et opérateurs. Elles ne sont plus optimales, rien n'assure même qu'elles fournissent un résultat intéressant dès qu'il y a des non linéarités présentes dans les équations. Or les équations régissant l'océan et l'atmosphère sont loin d'être linéaires. Les erreurs de diverses natures n'étant pas

toujours bien connues, elles viennent s'ajouter à des calculs déjà approximatifs du fait des linéarisations qui interviennent tout au long de ces méthodes.

Quelques pistes sont actuellement explorées pour essayer d'atténuer les problèmes liés aux erreurs mal identifiées ainsi qu'aux non linéarités. Il faut notamment citer la classe d'algorithmes de type 4D-PSAS, qui permettent déjà de supprimer le problème de l'erreur modèle. De plus, il existe désormais un certain recul dans le monde de la prévision qui permet d'avoir de bonnes statistiques d'erreurs (sur l'ébauche et sur les observations).

L'enjeu est évidemment d'améliorer sensiblement les résultats de l'assimilation, et par là-même, les prévisions à plus ou moins court terme. Celles-ci ont déjà fait d'énormes progrès, et les prévisions météo à 3 jours sont du même ordre de fiabilité que les prévisions à 24 heures d'il y a quelques années. L'objectif est maintenant d'obtenir des prévisions que l'on pourrait qualifier de très fiables (faible taux d'erreur) une semaine à l'avance en météo.

Une seconde difficulté de ces méthodes est leur coût de mise en œuvre numérique exorbitant. Ceci vient de la dimension du problème, environ 10^6 à 10^7 . À ce stade, il devient délicat de travailler avec des matrices, et une simple résolution d'un système d'équations différentielles peut déjà s'avérer coûteuse.

Là encore, plusieurs pistes sont explorées : le filtre SEEK avec la réduction d'ordre pour les méthodes séquentielles ; le 4D-VAR séquentiel pour réduire là aussi la dimension du vecteur de contrôle, le 4D-PSAS qui se place dans l'espace des observations, de dimension plus petite, pour les méthodes variationnelles.

L'objectif de réduire les dimensions du problème est double. D'une part on peut améliorer les modèles (plus de points de grille par exemple) et les résultats (minimisation plus poussée dans les méthodes variationnelles), mais on doit aussi rendre ces méthodes opérationnelles. En effet, il faut pouvoir traiter les observations que l'on récupère dans un certain laps de temps en beaucoup moins de temps. En imaginant qu'on récupère en un seul coup toutes les observations réalisées sur une fenêtre de 24 heures, il faudra réussir à appliquer une méthode d'assimilation de données les utilisant en beaucoup moins que 24 heures. Sinon le simple fait de traiter ces observations aura fait perdre 24 heures de prévision opérationnelle. Si les prévisions ne sont très fiables qu'à 48 heures, on ne pourra fournir des prévisions utilisables que pour 24 heures. L'enjeu est donc énorme puisque chaque heure gagnée sur l'analyse des données est une heure de prévision utilisable gagnée.

2.3.2 Projets opérationnels

De très nombreux projets sont partis de cette nécessité d'avoir des modèles opérationnels. En voici quelques-uns qui sont actuellement en développement en vue de remplacer ceux utilisés dans les différents organismes de prévision opérationnelle.

MERCATOR

Le projet MERCATOR, projet français, a pour but de développer un modèle océanographique opérationnel. Il doit permettre de décrire à tout instant l'état complet (près des côtes et au large, en surface et en profondeur) de tous les océans du monde. Le modèle utilisé devra simuler l'océan global en reposant sur les équations primitives à haute résolution. En vue de prévisions opérationnelles, le modèle doit être capable d'assimiler les observations satellitaires et in situ de l'océan.

CLIPPER

Le projet CLIPPER consiste en une modélisation à haute résolution de la circulation dans l'océan Atlantique, forcée et couplée avec l'atmosphère. Il met en place des configurations à basse, moyenne et haute résolution et s'intègre dans le projet mondial WOCE (World Ocean Circulation Experiment) qui cherche à acquérir une base de données représentative de l'état actuel de l'océan et décrire sa circulation générale.

GODAE

Le projet GODAE (Global Ocean Data Assimilation Experiment) est un projet mondial qui doit permettre d'ici 2 ans de développer un modèle d'assimilation opérationnelle sur un océan global. À terme, il regroupera plusieurs projets développés dans différents centres d'océanographie opérationnelle (MERCATOR en fait partie) afin d'obtenir conjointement des prévisions opérationnelles dans ces différents centres.

2.3.3 Objectifs

Notre objectif est d'étudier quelques méthodes d'assimilation de données jusque là peu utilisées, essentiellement la méthode duale. Nous avons testé le 4D-VAR et sa version duale sur un océan quasi-géostrophique barocline afin de tester la faisabilité d'un modèle de prévisions océaniques opérationnelles, basé sur le 4D-PSAS. En effet, à l'heure actuelle, de nombreux centres de prévision météo (Météo France, ECMWF, ...) ont choisi de travailler avec des méthodes variationnelles. Il convient donc de poursuivre dans ces méthodes, et le 4D-PSAS est, sur le plan théorique, une alternative très intéressante au 4D-VAR qu'il faut désormais tester numériquement.

Chapitre 3

Minimisation

Ce travail a été réalisé en collaboration avec Fabrice Veersé et a conduit à la publication de [51] (qui se trouve en annexe) et [50].

3.1 Introduction

En météorologie et en océanographie, les prévisions opérationnelles reposent largement sur une bonne estimation de l'état initial à partir duquel sera calculée l'évolution future du système. La reconstitution de cet état initial peut être réalisée en utilisant les observations disponibles grâce à des techniques d'assimilation variationnelle de données qui consistent à calculer la meilleure estimation possible de l'état initial grâce à une minimisation explicite de la fonctionnelle qui mesure l'écart entre les observations et les états estimés du système.

La minimisation de la fonctionnelle s'effectue assez difficilement (fonctionnelle non convexe) à l'aide de méthodes de type quasi-Newton, qui diffèrent de la méthode de Newton par l'utilisation d'approximations successives de la hessienne inverse de la fonctionnelle au point courant, approximations moins coûteuses à mettre en œuvre que la vraie hessienne. Néanmoins, de nombreux paramètres doivent être stockés afin d'approcher au mieux la hessienne. Pour des raisons de coût en mémoire, on utilise des algorithmes dits à *mémoire limitée* qui consistent à ne garder qu'une quantité finie, relativement faible, des informations nécessaires à la mise à jour de l'approximation de la hessienne.

Nous utiliserons ici l'algorithme de minimisation inverse BFGS (Broyden-Fletcher-Goldfarb-Shanno) à mémoire limitée. Au cours de la minimisation de la fonctionnelle, on construit une approximation de la hessienne inverse. À l'optimum, celle-ci représente une information importante pour le problème d'identification considéré, car elle représente les covariances d'erreur du minimum calculé. En pratique, pour des raisons de coût de calcul, la minimisation est arrêtée avant convergence. Le problème

essentiel est donc de savoir quelle est la qualité de l'approximation de la hessienne inverse fabriquée.

3.2 Algorithmes de descente

3.2.1 Algorithme de Newton

Supposons que l'on veuille minimiser une fonctionnelle J dépendant d'un paramètre x . L'algorithme de Newton consiste en la méthode suivante :

1. Initialisation : choix d'un point de départ de la minimisation x_0 .
2. Itérations :
 - À partir de x_k , on construit la direction de descente

$$d_k = -[\nabla^2 J(x_k)]^{-1} \cdot \nabla J(x_k)$$

qui minimise l'approximation quadratique locale

$$J(x) \approx J(x_k) + (x - x_k) \cdot \nabla J(x_k) + \frac{1}{2} (x - x_k)^T \cdot \nabla^2 J(x_k) \cdot (x - x_k).$$

- On calcule alors un pas de descente ρ_k le long de la direction de descente d_k par recherche linéaire (on minimise la fonctionnelle le long de d_k) de sorte que

$$J(x_k + \rho_k d_k) \leq J(x_k) + \alpha \rho_k [\nabla J(x_k)]^T \cdot d_k$$

et

$$|[\nabla J(x_k + \rho_k d_k)]^T \cdot d_k| \leq \beta |[\nabla J(x_k)]^T \cdot d_k|$$

avec $0 < \alpha < \beta < 1$ (conditions de Wolfe, [54]).

- On pose alors $x_{k+1} = x_k + \rho_k d_k$.

3. On réitère ce procédé jusqu'à convergence de la suite (x_k) vers le minimum de la fonctionnelle J .

Cet algorithme est très efficace, mais dans un cas d'assimilation de données opérationnelles sur un système complexe, il est très coûteux à mettre en œuvre puisqu'il nécessite à chaque étape le calcul de la hessienne inverse de la fonctionnelle au point considéré.

3.2.2 Algorithme de type quasi-Newton

L'idée de cet algorithme (voir [13]) est de remplacer la hessienne $H = \nabla^2 J$ (ou son inverse $W = [\nabla^2 J]^{-1}$) par une suite d'approximations symétriques définies positives, que l'on met à jour à chaque itération, pour un coût relativement faible. La mise

à jour de cette approximation lors d'une étape de l'algorithme est en général une correction de rang 2 : pour passer de l'étape k à l'étape $k + 1$, on pose

$$W_{k+1} = W_k + \alpha u \cdot u^T + \beta v \cdot v^T$$

où α et β sont des scalaires, et u et v des vecteurs.

L'algorithme est donc le suivant :

- si on a à l'étape k une approximation W_k de $[\nabla^2 J(x_k)]^{-1}$, on définit la direction de descente

$$d_k = -W_k \cdot \nabla J(x_k),$$

- on cherche un pas de descente ρ_k par recherche linéaire,
 - on pose $x_{k+1} = x_k + \rho_k d_k$,
 - puis on calcule une approximation W_{k+1} de $[\nabla^2 J(x_{k+1})]^{-1}$ avec une formule de mise à jour de rang 2 utilisant uniquement x_k , x_{k+1} , $\nabla J(x_k)$ et $\nabla J(x_{k+1})$.
- Toute la difficulté consiste alors à trouver une bonne formule de mise à jour.

3.2.3 Algorithme BFGS

L'algorithme BFGS (voir [7]), dû à Broyden, Fletcher, Goldfarb et Shanno, est un algorithme de type quasi-Newton où la formule de mise à jour de l'approximation de la hessienne inverse est :

$$W_{k+1} = U(W_k, s_k, y_k) = \left(I - \frac{s_k \otimes y_k}{\langle y_k, s_k \rangle} \right) W_k \left(I - \frac{y_k \otimes s_k}{\langle y_k, s_k \rangle} \right) + \frac{s_k \otimes s_k}{\langle y_k, s_k \rangle} \quad (3.1)$$

avec $s_k = x_{k+1} - x_k$, $y_k = \nabla J(x_{k+1}) - \nabla J(x_k)$, et $u \otimes v : d \rightarrow \langle v, d \rangle u$.

En pratique, on stocke W_0 et les paires (s_k, y_k) calculées à chaque étape. En effet, il est trop coûteux de stocker toutes les matrices W_k . L'inconvénient majeur de cet algorithme est le coût de stockage de toutes les paires (s_k, y_k) , surtout lorsque la dimension du problème est importante (de l'ordre de 10^6).

3.2.4 Algorithme L-BFGS

Cet algorithme ([30], [39]) est une extension de l'algorithme précédent, à mémoire limitée. Concrètement, on se donne un entier M , de l'ordre de 5, et on ne stocke que les M dernières paires (s, y) calculées. La mise à jour des matrices W_k se fait alors de la manière suivante :

$$\begin{aligned}
0 \leq k \leq M-1 : \quad & W_{k+1} = U(W_k, s_k, y_k), \\
k \geq M+1 : \quad & \begin{cases} W_k^0 = D_k, \\ W_k^{j+1} = U(W_k^j, s_{k-M+j}, y_{k-M+j}), \quad 0 \leq j \leq M-1 \\ W_k = W_k^M \end{cases} \quad (3.2)
\end{aligned}$$

où D_k est une matrice diagonale.

3.2.5 Mise à jour de la matrice diagonale

Il existe plusieurs formules de mise à jour de la matrice diagonale D_k permettant une bonne initialisation de l'algorithme L-BFGS. Les trois premières proviennent de [19], et la dernière est une version généralisée de la formule de mise à jour de quasi-Cauchy [55].

BFGS inverse La $i^{\text{ème}}$ composante de la matrice diagonale mise à jour est :

$$D_+^{(i)} = D^{(i)} + \left(\frac{1}{\langle y, s \rangle} + \frac{\langle Dy, y \rangle}{\langle y, s \rangle^2} \right) \langle s, e_i \rangle^2 - \frac{2D^{(i)} \langle y, e_i \rangle \langle s, e_i \rangle}{\langle y, s \rangle}. \quad (3.3)$$

$(e_i)_{1 \leq i \leq n}$ désigne ici une base orthonormée de \mathbb{R}^n pour le produit scalaire $\langle \cdot, \cdot \rangle$.

BFGS directe La formule de mise à jour BFGS directe du préconditionneur diagonal D est la suivante :

$$D_+^{(i)} = \left(\frac{1}{D^{(i)}} + \frac{\langle y, e_i \rangle^2}{\langle y, s \rangle} - \frac{\langle s, e_i \rangle}{D^{(i)^2} \langle D^{-1} s, s \rangle} \right)^{-1}. \quad (3.4)$$

DFP inverse La formule de mise à jour DFP (Davidon-Fletcher-Powell) inverse est :

$$D_+^{(i)} = D^{(i)} + \frac{\langle s, e_i \rangle^2}{\langle y, s \rangle} - \frac{(D^{(i)} \langle y, e_i \rangle)^2}{\langle Dy, y \rangle}. \quad (3.5)$$

Quasi-Cauchy La formule de mise à jour quasi-Cauchy peut s'écrire de la façon suivante :

$$D_+ = \begin{cases} D & \text{si } \langle Dy, y \rangle = \langle y, s \rangle \\ (I + \nu G)^{-2} D & \text{si } \langle Dy, y \rangle \neq \langle y, s \rangle \end{cases} \quad (3.6)$$

où G est une matrice diagonale dont la $i^{\text{ème}}$ composante est $\langle y, e_i \rangle^2$, et ν est la plus grande solution de l'équation $F(\nu) = \langle y, s \rangle$ avec

$$F(\nu) = \langle (I + \nu G)^{-2} D y, y \rangle.$$

Cette formule de mise à jour du préconditionneur diagonal est obtenue en résolvant le problème de minimisation

$$\min \langle w, w \rangle \text{ sous la contrainte } \langle (D^{\frac{1}{2}} + \Omega)^2 y, y \rangle = \langle y, s \rangle$$

où Ω est la matrice diagonale dont les éléments diagonaux sont les composantes du vecteur w .

3.2.6 Mise à l'échelle de la matrice diagonale

Si on effectue un développement limité de $\nabla J(x_{k+1})$ au voisinage de x_k , on obtient :

$$\begin{aligned} \nabla J(x_{k+1}) &= \nabla J(x_k) + \nabla^2 J(x_k) \cdot (x_{k+1} - x_k) + \dots \\ (i.e.) \quad y_k &\approx \nabla^2 J(x_k) \cdot s_k \end{aligned}$$

et donc, pour que W_k soit une bonne approximation de $[\nabla^2 J(x_k)]^{-1}$, on peut imposer à W_k de vérifier $W_k y_k = s_k$ le long de y_k , c'est-à-dire vérifier la relation de quasi-Cauchy :

$$\langle W_k y_k, y_k \rangle = \langle y_k, s_k \rangle. \quad (3.7)$$

Par conséquent, il faudra effectuer avant (ou éventuellement après) la mise à jour de la matrice diagonale une mise à l'échelle, de sorte que la nouvelle matrice diagonale ait la bonne propriété. Il conviendra donc de multiplier D par $\frac{\langle y, s \rangle}{\langle D y, y \rangle}$.

Cette mise à l'échelle permet d'économiser un certain nombre d'itérations et d'évaluations de la fonction coût et de son gradient [19].

L'impact de cette mise à jour sera étudié par la suite pour les trois premières formules de mise à jour du préconditionneur diagonal. En effet, cette mise à l'échelle n'a aucun sens dans le cas de la formule de quasi-Cauchy (3.6) puisque par construction, la nouvelle matrice diagonale vérifie déjà la condition (3.7).

3.3 Hessienne et covariances d'erreur

Étudions désormais le lien entre la matrice hessienne et la matrice de covariance des erreurs d'analyse. Si on note y les observations du système, x_e une estimation

de l'état initial du système et x_i l'état initial réel du système, on cherche alors à minimiser la fonctionnelle suivante :

$$\begin{aligned} J(x) &= J_e(x) + J_o(x) \\ &= \frac{1}{2}(x - x_e)^T P_e(t_0)^{-1}(x - x_e) + \frac{1}{2}(H(x) - y)^T R^{-1}(H(x) - y)^T, \end{aligned} \quad (3.8)$$

où H est un opérateur d'observation qui fournit une estimation des observations du système à l'état correspondant, $P_e(t_0)$ et R sont les matrices de covariance d'erreur relatives à l'état initial ($x_i - x_e$) et aux observations ($H(x_i) - y$) respectivement.

Le minimum x^* de la fonctionnelle J est caractérisé par

$$\begin{aligned} \nabla J(x^*) &= [P_e(t_0)]^{-1}(x^* - x_e) + H'(x^*)^T R^{-1}(H(x^*) - y) \\ &= 0. \end{aligned} \quad (3.9)$$

La matrice hessienne au minimum est alors :

$$\begin{aligned} \nabla^2 J(x^*) &= [P_e(t_0)]^{-1} + H'(x^*)^T R^{-1} H'(x^*) \\ &+ \frac{1}{2} \left[H''(x^*)^T R^{-1}(H(x^*) - y) + (H(x^*) - y)^T R^{-1} H''(x^*) \right] \end{aligned} \quad (3.10)$$

En supposant H linéaire et en introduisant l'état initial réel du système x_i dans l'équation (3.9), on obtient :

$$[P_e(t_0)]^{-1}((x^* - x_i) - (x_e - x_i)) + H'(x^*)^T R^{-1}(H(x_i) + H'(x_i)(x^* - x_i) - y) = 0,$$

soit encore

$$\begin{aligned} ([P_e(t_0)]^{-1} + H'(x^*)^T R^{-1} H'(x_i))(x^* - x_i) &= [P_e(t_0)]^{-1}(x_e - x_i) \\ &- H'(x^*)^T R^{-1}(H(x_i) - y). \end{aligned}$$

En multipliant à droite par l'expression transposée et en prenant l'espérance mathématique, on trouve la matrice des covariances d'erreur d'analyse :

$$P^* = E((x^* - x_i)(x^* - x_i)^T) = ([P_e(t_0)]^{-1} + H'(x^*)^T R^{-1} H'(x_i))^{-1}, \quad (3.11)$$

sous réserve d'avoir les hypothèses suivantes :

- l'erreur relative à la condition initiale est d'espérance nulle et de variance R ,
- l'erreur relative aux observations est d'espérance nulle et de variance $P_e(t_0)$,
- ces deux erreurs sont décorrélées.

On peut toutefois remarquer que la formule (3.11) donnant la matrice des covariances d'erreur d'analyse P^* se déduit directement de (3.8) lorsque l'opérateur H est supposé linéaire.

On en déduit alors que la hessienne inverse approche la matrice des covariances d'erreur d'analyse lorsque les non linéarités sont négligeables et que x^* est dans un voisinage de x_i où H' peut être considérée comme constante.

3.4 Application à l'équation de Burgers

3.4.1 Équations du modèle

Nous nous plaçons ici dans le cas particulier d'un modèle régi par l'équation de Burgers 1D sur un cercle :

$$\frac{\partial x}{\partial t} + \frac{1}{2} \frac{\partial x^2}{\partial s} - \nu \frac{\partial^2 x}{\partial s^2} = 0 \quad (3.12)$$

où s représente l'abscisse curviligne le long du parallèle 45°N et ν est le coefficient de diffusion, choisi égal à $10^5 \text{ m}^2 \cdot \text{s}^{-1}$. Le domaine étant le 45^{ème} parallèle Nord, il peut être considéré comme cyclique, de période la circonférence du parallèle, soit environ $28.3 \times 10^6 \text{ m}$.

3.4.2 Paramètres des expériences numériques

Les observations sont réalisées à intervalles de temps réguliers, à des endroits fixés choisis aléatoirement. Les observations sont alors bruitées. Les erreurs d'observation suivent une loi normale, sont décorrélées et ont toutes la même variance, proportionnelle à la matrice identité.

Le maillage à *grande résolution* utilisé pour discrétiser le domaine comporte 258 points de grille, et la dimension du problème discrétisé est donc 258. Mais dans un premier temps, nous utiliserons une grille de discrétisation à *faible résolution*, avec seulement 130 points, et donc la dimension du vecteur de contrôle est restreinte à 130 afin de pouvoir effectuer un grand nombre d'itérations. Le nombre de paires (s, y) stockées est 5. Les paramètres de recherche linéaire de Wolfe sont $\alpha = 10^{-4}$ et $\beta = 0.9$. Le nombre maximal d'itérations a été fixé à 120, et le nombre maximal d'évaluations de la fonction coût et de son gradient est 144 (le nombre maximal d'itérations augmenté d'un cinquième de sa valeur). La décroissance attendue de la fonction coût à la première itération est la moitié de sa valeur. La fonctionnelle quadratique approchée que l'on souhaite minimiser est la suivante :

$$\tilde{J}(x) = \frac{1}{2}(x - x_e)^T [P_e(t_0)]^{-1} (x - x_e) + \frac{1}{2}(\tilde{H}(x) - y)^T R^{-1}(\tilde{H}(x) - y), \quad (3.13)$$

où \tilde{H} est une approximation linéaire de H , comprenant notamment la linéarisation du modèle. Le critère d'arrêt du processus de minimisation est le suivant :

$$\frac{\|\nabla \tilde{J}(x_k)\|}{\|\nabla \tilde{J}(x_0)\|} < \varepsilon_g$$

où $\|\cdot\|$ est la norme induite par le produit scalaire $\langle \cdot, \cdot \rangle$ et ε_g est de l'ordre de 1.5×10^{-8} . La minimisation est réalisée à l'aide du minimiseur M1QN3 de l'INRIA.

La qualité de l'approximation de la hessienne inverse L-BFGS est mesurée en calculant le spectre de

$$W_{true}^{-1} - W_{L-BFGS}^{-1} \quad (3.14)$$

où W_{true}^{-1} est la vraie hessienne, calculée à l'aide de la méthode de l'adjoint au second ordre, et W_{L-BFGS}^{-1} est l'approximation L-BFGS de la hessienne construite lors de la minimisation.

Afin de pouvoir mesurer aussi de manière relative cette qualité, le spectre de

$$I - W_{true}^{-1} W_{L-BFGS} \quad (3.15)$$

est aussi calculé. Comme W_{true}^{-1} et W_{L-BFGS} sont symétriques, le spectre de $I - W_{L-BFGS} W_{true}^{-1}$ est sensiblement équivalent à celui de (3.15), et nous nous limiterons à calculer le spectre de $I - W_{true}^{-1} W_{L-BFGS}$.

Dans les deux cas, le spectre est calculé en utilisant une méthode de type Arnoldi [26].

Même si nous nous sommes essentiellement intéressés à la qualité de l'approximation de la hessienne L-BFGS, nous avons également regardé l'efficacité de l'algorithme de minimisation en mesurant le nombre d'itérations et de simulations (évaluations de la fonction coût et de son gradient) nécessaires à la convergence.

3.4.3 Convergence de l'algorithme BFGS à mémoire illimitée

Avant d'étudier le cas de l'algorithme BFGS à mémoire limitée, nous avons vérifié la convergence de l'algorithme BFGS. Ceci a été réalisé en prenant une valeur de M (le nombre de paires stockées) plus grande que le nombre d'itérations utilisées.

La figure 3.1 montre les spectres de (3.14) et (3.15) pour plusieurs nombres maximaux d'itérations. La convergence est assez claire.

3.4.4 Choix des paires stockées

Il reste un degré de liberté dans l'algorithme L-BFGS, celui du choix des paires (s, y) stockées. L'algorithme (3.2) montre que le préconditionneur diagonal doit ressembler autant que possible à W_{k-M-1} . Ceci suggère d'utiliser la paire qui est sur le point d'être effacée, la *plus vieille* paire, (s_{k-M}, y_{k-M}) pour mettre à jour la matrice diagonale D_k . Pour les M premières itérations, la matrice diagonale initiale doit être utilisée. Ceci est en accord avec le fait que toute l'information provenant des M dernières itérations est entièrement stockées dans les paires (s, y) . Néanmoins, il est envisageable d'utiliser pour la mise à jour du préconditionneur diagonal la dernière paire (s, y) construite, la paire la *plus récente*. Cela va donc avoir tendance à rajouter du poids sur l'information récemment construite.

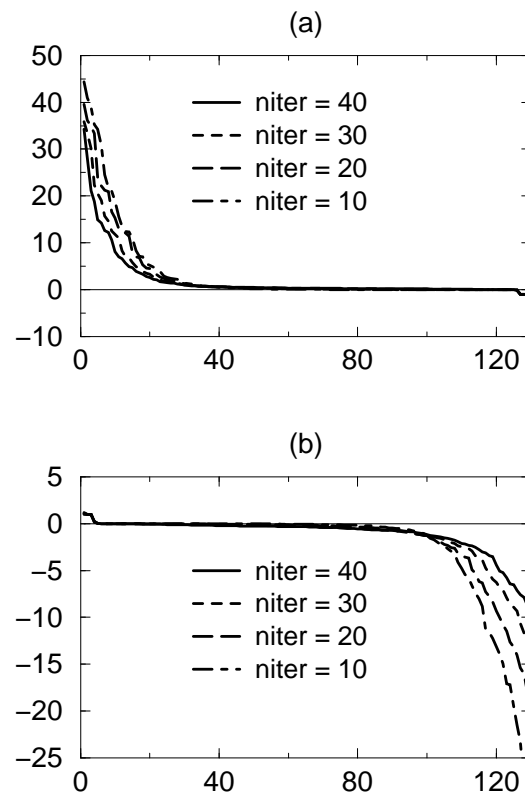


FIG. 3.1 – Spectre des opérateurs pour différents nombres d'itérations de l'algorithme de minimisation BFGS. Différence des hessiennes $W_{true}^{-1} - W_{BFGS}^{-1}$ (a) et différence relative $I - W_{true}^{-1} W_{BFGS}^{-1}$ (b).

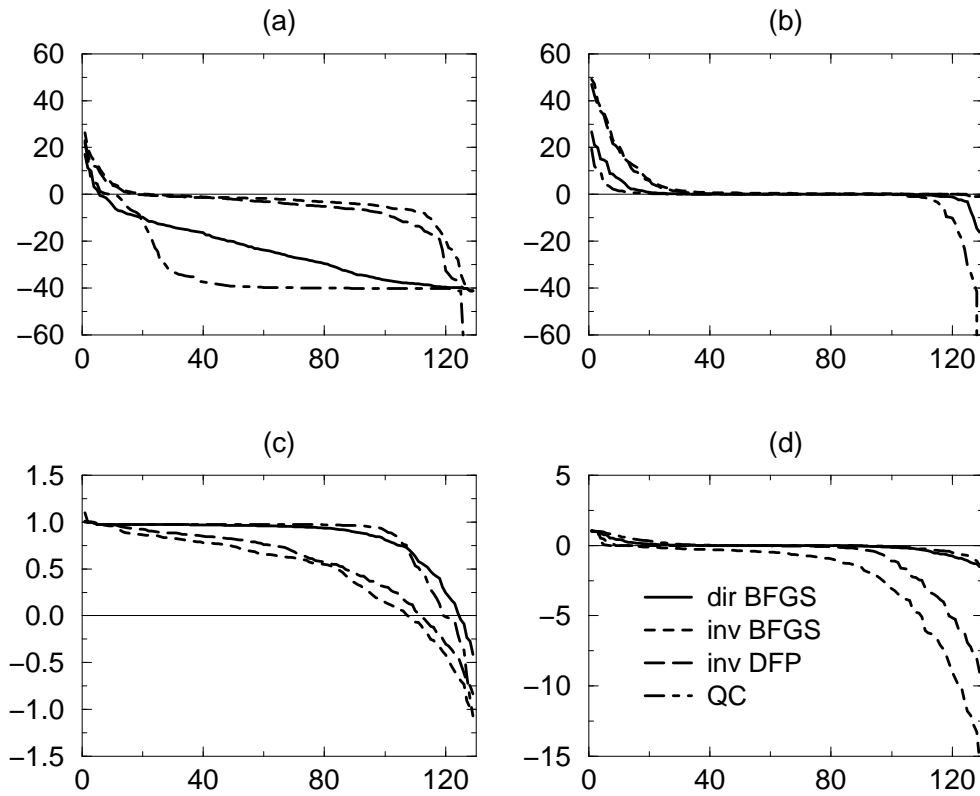


FIG. 3.2 – Spectre de l'opérateur $W_{true}^{-1} - W_{L-BFGS}^{-1}$ pour les différentes formules de mise à jour en utilisant les paires les plus vieilles (a) et les plus récentes (b) respectivement. Spectre de l'opérateur $I - W_{true}^{-1} W_{L-BFGS}$ pour les différentes formules de mise à jour en utilisant les paires les plus vieilles (c) et les plus récentes (d) respectivement.

Formule	Paire la plus vieille	Paire la plus récente
BFGS directe	77/78	40/52
BFGS inverse	69/70	42/84
DFP inverse	68/69	42/84
Quasi-Cauchy	96/103	120/130

TAB. 3.1 – Nombre d'itérations/simulations nécessaires à la convergence pour les différentes formules de mise à jour sans mise à l'échelle, en utilisant la paire la plus vieille ou la plus récente.

Pour étudier la différence entre ces deux façons de stocker les M paires, les formules de mise à jour sont utilisées sans mise à l'échelle. Les spectres des deux opérateurs de comparaison des hessiennes sont représentés sur la figure 3.2. Il est clair que l'utilisation des paires les plus récentes fournit une approximation de la hessienne bien meilleure qu'en utilisant les paires les plus anciennes.

Le tableau 3.1 montre les nombres d'itérations et de simulations correspondantes nécessaires à la convergence. À part pour la formule BFGS directe, l'utilisation de la paire la plus récente augmente le nombre de simulations, même si cela diminue d'un autre côté le nombre d'itérations. La formule de quasi-Cauchy a des résultats globalement moins bons que les trois autres formules de mise à jour. L'utilisation de la paire la plus récente avec la formule de mise à jour BFGS directe semble donner à la fois une bonne approximation de la hessienne et accélérer la convergence.

3.4.5 Impact de la mise à l'échelle

Gilbert et Lemaréchal [19] ont montré que la mise à l'échelle de la matrice diagonale, permettant la relation de quasi-Cauchy (3.7), rendait la minimisation plus performante. Comme la mise à l'échelle n'a aucun sens pour la formule de mise à jour de quasi-Cauchy, son impact n'a été testé que pour les trois premières formules de mise à jour.

La figure 3.3 montre, pour les trois formules de mise à jour, les spectres des deux opérateurs de comparaison lorsque la mise à l'échelle a lieu avant ou après la mise à jour. Une comparaison avec la figure 3.2 montre une détérioration certaine de la qualité de l'approximation de la hessienne construite lors de la minimisation.

Le tableau 3.2 montre les valeurs correspondantes du nombre d'itérations/simulations nécessaires. La mise à l'échelle conduit à une diminution du nombre de simulations, c'est-à-dire du nombre d'évaluations de la fonction coût et de son gradient. Les formules de mise à jour BFGS directe et DFP inverse se comportent sensiblement de

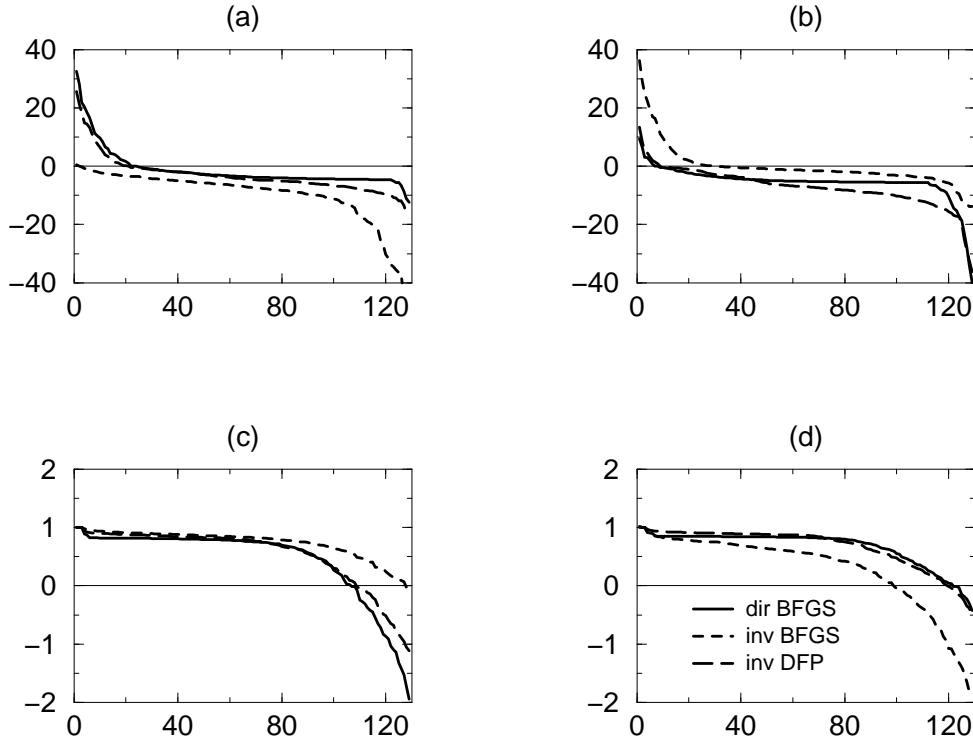


FIG. 3.3 – Spectre de l’opérateur $W_{true}^{-1} - W_{L-BFGS}^{-1}$ lorsque la mise à l’échelle de la matrice diagonale a lieu avant (a) et après (b) sa mise à jour. Spectre de l’opérateur $I - W_{true}^{-1} W_{L-BFGS}$ lorsque la mise à l’échelle du préconditionneur diagonal a lieu avant (c) et après (d) sa mise à jour.

Formule	Mise à l’échelle avant	Mise à l’échelle après
BFGS directe	47/49	47/49
BFGS inverse	55/60	51/53
DFP inverse	52/53	50/52

TAB. 3.2 – Nombre d’itérations/simulations nécessaires à la convergence pour les différentes formules lorsque la mise à l’échelle a lieu avant ou après sa mise à jour.

Formule	Nombre d'itérations/simulations
BFGS directe	40/43
BFGS inverse	44/46
DFP inverse	43/46

TAB. 3.3 – Nombre d'itérations/simulations nécessaires à la convergence pour les différentes formules lorsque la mise à l'échelle a lieu uniquement pour la minimisation et pas pour la mise à jour.

la même façon suivant que la mise à l'échelle est faite avant ou après la mise à jour, mais la formule BFGS inverse donne tout de même de meilleurs résultats lorsque la mise à l'échelle a lieu après.

3.4.6 Nouvelle approche de la mise à l'échelle

Afin d'obtenir une bonne approximation de la hessienne, il est clairement plus approprié d'utiliser la paire la plus récente pour mettre à jour le préconditionneur. Il est également préférable, du point de vue du nombre d'itérations/simulations nécessaires à la convergence de la minimisation, de le mettre à l'échelle après sa mise à jour. Mais cela a tendance à détériorer nettement l'approximation de la hessienne.

Il est alors naturel d'envisager une nouvelle façon de mettre à jour le préconditionneur diagonal, dans laquelle la matrice serait mise à jour sans mise à l'échelle, mais sa version mise à l'échelle serait utilisée pour la minimisation. A priori, cela devrait accélérer la convergence de la minimisation (grâce à la mise à l'échelle) sans détériorer la qualité de l'approximation de la hessienne (puisque l'on utilise la version non mise à l'échelle pour la mise à jour).

La figure 3.4 montre les spectres respectifs des opérateurs de différence entre les hessiennes calculée et exacte dans ce cas. On retrouve alors des spectres comparables aux figures 3.2-b et -d, lorsque l'approximation de la hessienne était très bonne.

Le tableau 3.3 donne les nombres d'itérations et simulations nécessaires à la convergence de l'algorithme de minimisation dans cette nouvelle approche. On constate alors que cette approche apporte une très nette amélioration de la convergence aux trois formules de mise à jour, et tout spécialement à la formule BFGS directe.

3.4.7 Choix de M

Nous nous intéressons désormais à l'influence du choix de M , le nombre de paires (s, y) stockées, sur la performance de l'algorithme de minimisation et la qualité de

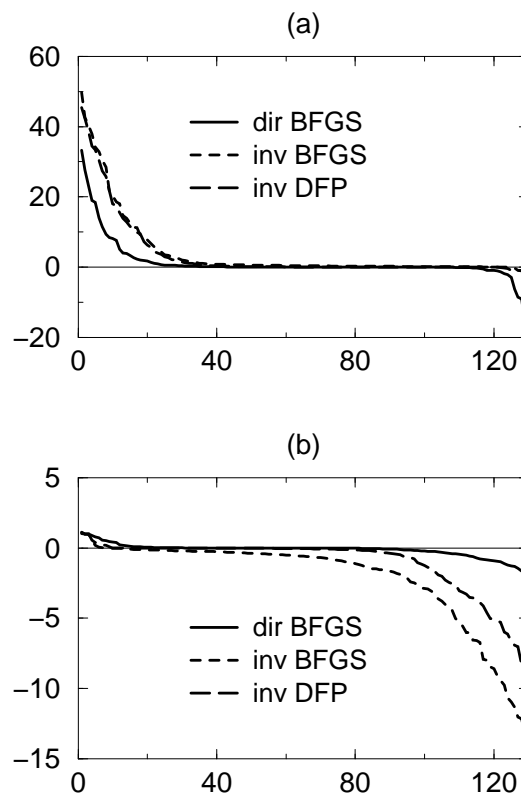


FIG. 3.4 – Spectres de $W_{true}^{-1} - W_{L-BFGS}^{-1}$ (a) et $I - W_{true}^{-1} W_{L-BFGS}$ (b) lorsqu'une version mise à l'échelle du préconditionneur est utilisée pour la minimisation et la version sans mise à l'échelle pour la mise à jour.

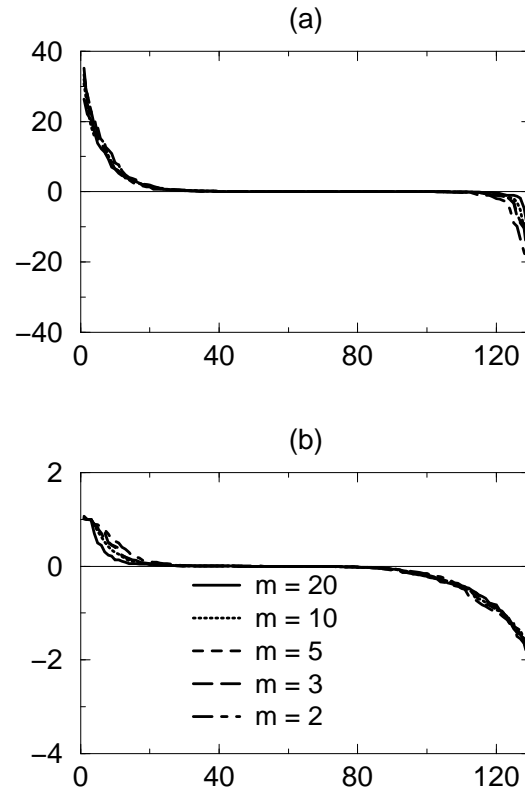


FIG. 3.5 – Spectres de $W_{true}^{-1} - W_{L-BFGS}^{-1}$ (a) et $I - W_{true}^{-1} W_{L-BFGS}$ (b) lorsqu'une version mise à l'échelle du préconditionneur est utilisée pour la minimisation et la version sans mise à l'échelle pour la mise à jour, pour différentes valeurs de M .

l'approximation de la hessienne construite. Nous nous limitons à la configuration qui a donné jusqu'à présent les meilleurs résultats, c'est-à-dire en utilisant la formule de mise à jour BFGS directe, en utilisant la paire la plus récente, et en ne mettant à l'échelle le préconditionneur que pour la minimisation et pas pour sa mise à jour.

La figure 3.5 montre l'évolution des spectres des deux opérateurs de comparaison pour différentes valeurs de M , allant de 2 à 20. L'impact sur la qualité de l'approximation de la hessienne est faible, même si plus M est grand, meilleure est l'approximation.

Le tableau 3.4 montre les nombres d'itérations/simulations correspondants pour les différentes valeurs de M . Là encore, plus M est choisi grand, plus l'algorithme de minimisation converge rapidement. La valeur de M est souvent choisie entre 3 et 20. En effet, une valeur plus faible entraînerait une dégradation relative des approximations construites ainsi que de la vitesse de convergence de l'algorithme de minimisa-

M	Nombre d'itérations/simulations
2	44/48
3	42/46
5	40/43
10	36/38
20	35/37

TAB. 3.4 – Nombre d'itérations/simulations nécessaires à la convergence de la minimisation en fonction du nombre M de paires stockées, en utilisant la formule BFGS directe, la paire la plus récente et la mise à l'échelle uniquement pour la minimisation.

tion, et une valeur plus grande reviendrait à travailler directement avec l'algorithme BFGS à mémoire illimitée, ce qui s'avère trop coûteux au niveau du stockage.

3.4.8 Évolution pendant la minimisation

Très souvent, pour des raisons de temps de calculs très élevés, la minimisation est arrêtée avant convergence dans les problèmes d'assimilation de données. Généralement, la minimisation est arrêtée après quelques dizaines d'itérations. Nous allons donc regarder l'évolution de la qualité de l'approximation de la hessienne construite en fonction du nombre d'itérations.

La figure 3.6 montre l'évolution correspondante dans le meilleur cas, c'est-à-dire la formule BFGS directe en utilisant la paire la plus récente et la nouvelle approche de la mise à l'échelle. Le nombre de paires stockées a de nouveau été fixé à 5. Il est assez évident que la qualité de l'approximation s'améliore au fur et à mesure que la minimisation est poussée. Ceci était prévisible car la dimension du sous-espace dans lequel est cherchée l'approximation augmente à chaque itération, alors que la vraie hessienne reste constante.

3.4.9 Cas non quadratique

Dans ce paragraphe, on s'intéresse à la minimisation de la fonctionnelle non quadratique J (et non \tilde{J}), lorsque l'opérateur d'observation et le modèle ne sont plus forcément linéaires. Du même coup, la hessienne n'est plus constante. De plus, la dimension du vecteur de contrôle est désormais égale à 258, soit la dimension de l'espace du modèle. La qualité de l'approximation de la hessienne est toujours vérifiée à l'aide de la méthode de l'adjoint au second ordre. Seuls les nombres maximaux d'itérations

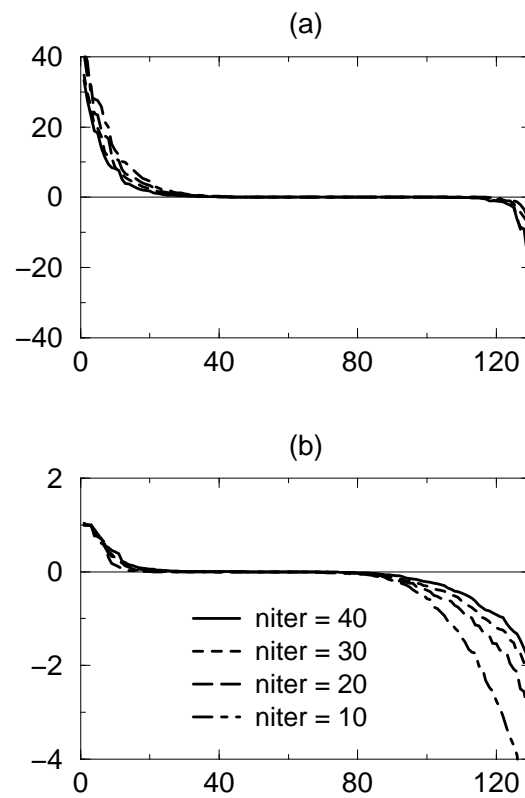


FIG. 3.6 – Spectres de $W_{true}^{-1} - W_{L-BFGS}^{-1}$ (a) et $I - W_{true}^{-1} W_{L-BFGS}$ (b) pour différentes valeurs du nombre maximal d'itérations.

Formule	Sans mise à l'échelle	
	Paire la plus vieille	Paire la plus récente
BFGS directe	78/79	-
BFGS inverse	76/77	-
DFP inverse	74/75	-
Quasi-Cauchy	98/139	188/248

Formule	Mise à l'échelle avant	Mise à l'échelle après	Nouvelle approche
	Paire la plus récente	Paire la plus récente	Paire la plus récente
BFGS directe	52/56	47/49	43/45
BFGS inverse	63/67	69/71	48/52
DFP inverse	55/57	56/58	48/51

TAB. 3.5 – Nombre d'itérations/simulations nécessaires à la convergence de la minimisation pour les différentes formules de mise à jour et dans les différents cas de figure rencontrés.

et simulations ont été changés, et valent désormais 200 et 250 respectivement. Les résultats ne diffèrent alors pratiquement pas du cas quadratique.

La seule différence est l'échec des trois formules BFGS directe, BFGS inverse et DFP inverse lorsque la paire la plus récente est utilisée pour la mise à jour, et qu'aucune mise à l'échelle n'est effectuée.

Le tableau 3.5 récapitule les nombres d'itérations et simulations nécessaires à la convergence de la minimisation pour les différentes formules et dans les différentes approches. Il faut remarquer que la formule de mise à jour de quasi-Cauchy a, à chaque fois, atteint le nombre maximal d'itérations ou de simulations avant la convergence de la minimisation.

Il faut en général plus d'itérations/simulations pour parvenir à la convergence de l'algorithme de minimisation que dans le cas quadratique simplifié. Néanmoins, la même conclusion reste valable, nous avons intérêt à utiliser la formule de mise à jour BFGS directe, en utilisant la paire la plus récente et la nouvelle approche pour la mise à l'échelle.

3.4.10 Conclusions relatives à ce problème test

Nous avons utilisé un problème test relativement simple, l'équation de Burgers en dimension 1, pour étudier les différentes formules de mise à jour de l'algorithme BFGS à mémoire limitée et les différentes stratégies qui s'offrent à nous. Nous nous sommes intéressés à la fois à la qualité de l'approximation de la hessienne construite, puisqu'elle correspond à une approximation de la matrice des covariances d'erreurs d'analyse, et à la convergence de l'algorithme de minimisation. La qualité de l'approximation de la hessienne a été mesurée à l'aide de la vraie hessienne, calculée avec la méthode de l'adjoint au second ordre. Il est évident que cela n'est numériquement réalisable que pour des problèmes de petite dimension.

Nous avons constaté plusieurs choses. Tout d'abord, l'utilisation de la paire (s, y) la plus récente donne de meilleurs résultats qu'avec la plus ancienne, sauf dans le cas non linéaire et sans mise à l'échelle. Ensuite, la formule de mise à jour de quasi-Cauchy a donné à de nombreuses reprises des résultats moins satisfaisants que les autres formules, aussi bien en terme de qualité de l'approximation de la hessienne qu'en convergence de la minimisation. La mise à l'échelle du préconditionneur diagonal accélère la convergence vers le minimum mais a tendance à détériorer la qualité de l'approximation de la hessienne. Enfin, il est possible d'accélérer la minimisation en mettant à l'échelle le préconditionneur diagonal, tout en ne détériorant pas l'approximation de la hessienne construite en utilisant une version non mise à l'échelle pour la mise à jour.

3.5 Conclusion

L'intérêt de la minimisation de la fonction coût est double, comme nous avons pu le voir. En effet, il y a la convergence (et la vitesse de convergence) de l'algorithme de minimisation vers le minimum de la fonctionnelle, mais il y a aussi la qualité de l'approximation de la hessienne construite lors de l'algorithme L-BFGS, puisque celle-ci est une très bonne approximation de la matrice de covariance d'erreur d'analyse, et elle a donc une importance statistique.

Nous avons vu plusieurs algorithmes de minimisation d'une fonction-coût, tous basés sur l'algorithme de Newton. Nous nous sommes plus particulièrement intéressés à l'algorithme L-BFGS, à mémoire limitée. Celui-ci est en effet couramment utilisé dans les problèmes d'assimilation variationnelle de données de grande taille, notamment en océanographie et météorologie. Il existe plusieurs degrés de liberté dans cet algorithme, notamment le choix de la formule de mise à jour, le nombre et le choix des paires de vecteurs stockées, l'utilisation ou non d'une mise à l'échelle du préconditionneur diagonal... Néanmoins, il convient de tester les différentes possibilités qui s'offrent à nous pour chaque problème test, car les différentes méthodes sont

loin de donner de façon universelle de bons ou mauvais résultats, comme on peut le voir dans [50]. Il y a certainement beaucoup d'autres façons d'utiliser les formules de l'algorithme L-BFGS, et il est également probable que pour certains problèmes, il existe de meilleures formules de mise à jour du préconditionneur diagonal.

Chapitre 4

Quasi-réversibilité et nudging

Dans ce chapitre, nous allons nous intéresser à deux grandes classes de méthodes rétrogrades, tout d'abord la quasi-réversibilité, puis le nudging. En effet, dans l'optique de l'identification d'un état initial, il peut sembler judicieux de traiter le système de façon rétrograde en temps, en remontant d'une condition finale.

4.1 Introduction sur la quasi-réversibilité

La méthode de quasi-réversibilité (Q.R.) a été introduite par R. Lattès et J.-L. Lions [24]. Elle consiste à modifier légèrement les problèmes rétrogrades pour les rendre bien posés (et numériquement résolubles). En supposant que l'état final d'un système soit relativement bien connu, une méthode assez naïve pour reconstruire l'état initial consiste à intégrer de façon rétrograde en temps l'équation d'évolution en partant de la condition finale. Malheureusement, les problèmes rétrogrades sont souvent mal posés et cela s'avère numériquement impossible. La méthode Q.R. autorise une intégration rétrograde et par conséquent permet d'obtenir numériquement une estimation de l'état initial à identifier ([23]).

Nous allons dans un premier temps voir le principe général de la méthode de quasi-réversibilité, puis l'appliquer de différentes façons à l'équation de la chaleur, puis au système de Lorenz.

4.2 Présentation de la méthode

4.2.1 Heuristique

Considérons un problème d'évolution bien posé (ceci sera assuré par certaines hypothèses que nous verrons plus loin) du type :

$$\begin{aligned} \frac{du}{dt} + A(t)u &= f, & 0 \leq t \leq T, \\ u(0) &= \xi, \end{aligned} \tag{4.1}$$

et cherchons la condition initiale ξ telle que la solution u de (4.1) vérifie la condition $u(T) = \chi$, χ étant la condition finale que l'on s'impose.

Généralement, le problème :

$$\begin{aligned} \frac{du}{dt} + A(t)u &= f, & 0 \leq t \leq T, \\ u(T) &= \chi \end{aligned} \tag{4.2}$$

est mal posé et ne permet donc pas de résoudre notre problème.

Par contre, sous certaines hypothèses (à définir), le problème

$$\begin{aligned} \frac{du_\varepsilon}{dt} + A(t)u_\varepsilon - \varepsilon A^*(t)A(t)u_\varepsilon &= f, & 0 \leq t \leq T, \\ u_\varepsilon(T) &= \chi \end{aligned} \tag{4.3}$$

admet une unique solution u_ε . Il est alors possible de définir une suite de problèmes perturbés

$$\begin{aligned} \frac{dU_\varepsilon}{dt} + A(t)U_\varepsilon &= f, & 0 \leq t \leq T, \\ U_\varepsilon(0) &= u_\varepsilon(0), \end{aligned} \tag{4.4}$$

et il convient de vérifier que $U_\varepsilon(T) \rightarrow \chi$ lorsque $\varepsilon \rightarrow 0$. Il ne restera qu'à choisir numériquement ε pour avoir la précision voulue.

4.2.2 Justifications théoriques

Soient V et H deux espaces de Hilbert avec $V \subset H$, V dense dans H . On note V' le dual de V et on identifie H à son dual, de sorte que $V \subset H \subset V'$. On se donne à chaque instant $t \in [0, T]$ un opérateur $A(t) \in \mathcal{L}(V; V')$. On peut alors construire une famille de formes bilinéaires $a(t; u, v)$ sur V .

Supposons que a vérifie :

1. $\forall u, v \in V$, la fonction $t \rightarrow a(t; u, v)$ est mesurable.

2. a est uniformément continue sur $[0, T] \times V \times V$:

$$\exists M > 0 \text{ tel que } \forall t \in [0, T], \forall u, v \in V, |a(t; u, v)|_H \leq M \|u\|_V \cdot \|v\|_V .$$

3. a vérifie la propriété de coercivité élargie :

$$\exists \lambda > 0, \alpha > 0 \text{ tels que } a(t; v, v) + \lambda \|v\|_H^2 \geq \alpha \|v\|_V^2, \forall t \in [0, T], \forall v \in V.$$

Alors nous avons les résultats suivants ([28]) :

Théorème 4.1 *Soient $f \in L^2([0, T], V')$ et $\xi \in H$. Sous les hypothèses énoncées précédemment sur a , il existe une unique solution u de (4.1) avec $u \in L^2([0, T], V)$ et $\frac{du}{dt} \in L^2([0, T], V')$. ■*

Notons $D(A(t))$ l'ensemble des fonctions acceptables en vue du problème (4.3) :

$$D(A(t)) = \{v \in V; A(t)v \in H\}.$$

Théorème 4.2 *Sous les hypothèses du théorème 4.1, en supposant en outre que $D(A(t))$ est un espace indépendant de t , alors le problème (4.3) admet une unique solution u_ε avec $u_\varepsilon \in L^2([0, T], D(A(t)))$ et $\frac{du_\varepsilon}{dt} \in L^2([0, T], D(A(t)))$. ■*

Démonstration : Voir [24]. □

Nous avons désormais la possibilité de résoudre le problème rétrograde perturbé. Intéressons nous à la convergence de la suite de solutions ainsi construites :

Théorème 4.3 *Sous les hypothèses du théorème 4.2, en supposant de plus que la forme bilinéaire a est indépendante du temps et symétrique (i.e. A auto-adjoint), alors $U_\varepsilon(T) \rightarrow \chi$ dans H lorsque $\varepsilon \rightarrow 0$, $U_\varepsilon \in L^2([0, T], V)$ étant défini par (4.4). ■*

Démonstration : Voir [24] et [6]. □

La convergence de la suite des états finaux $U_\varepsilon(T)$ vers l'état final de référence χ est acquise, mais la convergence de la suite U_ε en n'importe quel autre point de $[0, T]$ n'a généralement pas lieu d'être. Nous allons maintenant appliquer numériquement la méthode de la quasi-réversibilité à l'équation de la chaleur.

4.3 Application à l'équation de la chaleur

4.3.1 Résultats théoriques

Considérons un ouvert Ω de \mathbb{R}^n , et $u = u(x, t)$ la solution (unique) de

$$\begin{aligned} \frac{\partial u}{\partial t} - \nu \Delta u &= 0, & x \in \Omega, t \geq 0 \\ u(x, t) &= 0, & x \in \partial\Omega, t \geq 0 \\ u(x, 0) &= \xi(x), & x \in \Omega, \end{aligned} \quad (4.5)$$

où ν est le coefficient de diffusion, que nous prendrons égal à 1 pour simplifier, quitte à renommer les variables d'espace-temps x et t . Pour simplifier le problème, nous nous placerons dans la suite en dimension 1 sur l'ouvert $\Omega =]0, 1[$ et sur l'intervalle $[0, T]$ en temps.

L'instabilité de l'équation de la chaleur rétrograde (obtenue après un changement de variable $t \leftrightarrow T - t$)

$$\begin{aligned} \frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} &= 0, & x \in \Omega, t \in [0, T] \\ u(0, t) = u(1, t) &= 0, & t \in]0, T[\\ u(x, 0) &= \chi(x), & x \in \Omega, \end{aligned} \quad (4.6)$$

est connue :

Théorème 4.4 $\exists (u_n)$ une suite de solutions de $\frac{\partial u}{\partial t} + \Delta u = 0$ telle que $\|u_n(x, 0)\|_{L^2} \rightarrow 0$ et $\|u_n(x, T)\|_{L^2} \rightarrow \infty$ lorsque $n \rightarrow \infty$. ■

Il n'existe donc en général pas de solutions au problème de la chaleur rétrograde (4.6).

Nous allons considérer le problème suivant, en accord avec la méthode de la quasi-réversibilité :

$$\begin{aligned} \frac{\partial u_\varepsilon}{\partial t} + \frac{\partial^2 u_\varepsilon}{\partial x^2} + \varepsilon \frac{\partial^4 u_\varepsilon}{\partial x^4} &= 0, & x \in \Omega, t \in [0, T], \varepsilon > 0, \\ u_\varepsilon(x, t) &= 0, & x \in \partial\Omega, t \in]0, T[\\ \Delta u_\varepsilon(x, t) &= 0, & x \in \partial\Omega, t \in]0, T[\\ u_\varepsilon(x, 0) &= \chi(x), & x \in \Omega. \end{aligned} \quad (4.7)$$

Les théorèmes 4.2 et 4.3 s'appliquent dans ce cas particulier (l'opérateur laplacien vérifie bien toutes les hypothèses de ces théorèmes), et en résolvant uniquement des problèmes bien posés (existence et unicité de la solution), on peut construire des solutions de l'équation de la chaleur (4.5) aussi proches que l'on veut de χ à l'instant final.

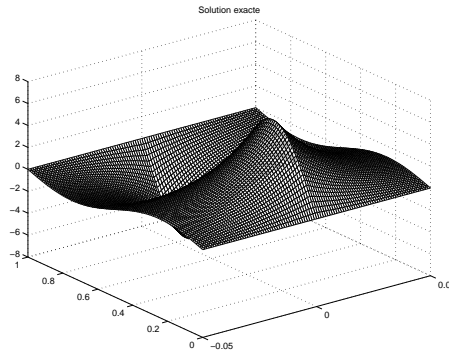


FIG. 4.1 – Solutions analytiques des problèmes (4.6) et (4.5), avec $T = 0.05$.

4.3.2 Expériences numériques

Les calculs ont été effectués avec le logiciel Matlab, en dimension 1, avec un domaine $\Omega =]0, 1[$, et avec $T = 0.05$ ou $T = 0.1$ selon les cas. Le pas d'espace a été fixé à 0.01. Le nombre de pas d'intégration en espace est égal à 50, quelle que soit la longueur de l'intervalle sur lequel nous avons travaillé. Le schéma d'intégration utilisé est Euler implicite, Crank-Nicolson ou Runge-Kutta d'ordre 4 suivant les cas et la précision voulue. La condition finale choisie est $\chi(x) = \sin(2\pi x)$.

Chacune des figures suivantes, concernant la quasi-réversibilité appliquée à l'équation de la chaleur, se présente sous la forme suivante : sur l'axe des abscisses, le temps t ; sur l'axe des ordonnées, la position x (graduée de 0.2 en 0.2); et suivant le dernier axe (vertical), pour les temps négatifs la solution du problème Q.R. (intégration rétrograde en partant de χ) puis, pour les temps positifs, la solution du problème direct (intégration directe de l'équation de la chaleur en partant de la solution finale du problème rétrograde). La qualité de la méthode se lit alors en comparant le premier instant (condition finale χ recherchée) et le dernier instant (approximation reconstruite de cette même condition finale). Lorsque ces deux états sont proches l'un de l'autre, cela signifiera que la méthode Q.R. nous aura permis de trouver un bon antécédent de χ par l'opérateur de la chaleur.

La figure 4.1 montre les solutions exactes (calculées de façon analytique) des équations rétrograde et directe de la chaleur. En séparant les variables temporelle et spatiale de l'équation (4.6), on peut calculer analytiquement la solution de l'équation de la chaleur rétrograde, puis celle de l'équation de la chaleur directe en partant de la solution précédente. Ces deux solutions sont les mêmes, à un changement de variable temporelle près, ce qui explique la symétrie de la figure par rapport au temps 0. On est alors certain que l'on va retomber sur χ au temps T . Néanmoins, même dans ces conditions particulières, il a été impossible d'intégrer numériquement l'équation de la chaleur rétrograde, comme nous le verrons également avec la méthode Q.R. dès que ε sera choisi trop petit.

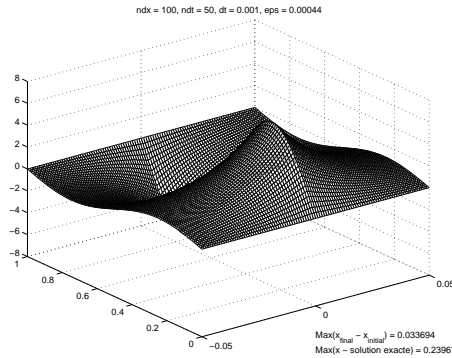


FIG. 4.2 – Solutions numériques des problèmes (4.7) et (4.5) avec $\varepsilon = 4.4 \cdot 10^{-4}$ et $T = 0.05$.

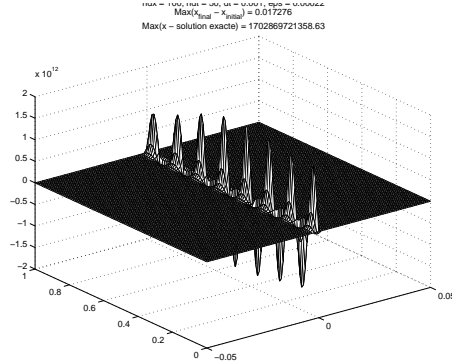


FIG. 4.3 – Solutions des problèmes (4.7) et (4.5) minimisant l'erreur pour $T = 0.05$, correspondant à $\varepsilon = 2.2 \cdot 10^{-4}$.

La figure 4.2 montre dans les temps négatifs la solution du problème quasi-réversible (4.7) avec $\varepsilon = 4.4 \cdot 10^{-4}$, puis dans les temps positifs la solution du problème direct (4.5) qui en découle. La valeur de ε a été choisie par dichotomie de sorte que la trajectoire ainsi construite suive au plus près la solution exacte tout au long de la période d'intégration. L'erreur en norme infinie entre la condition finale attendue χ et la condition finale reconstruite $U_\varepsilon(T)$ est de l'ordre de 3%. On peut alors considérer que la quasi-réversibilité permet bien d'intégrer de façon rétrograde l'équation de la chaleur pour certaines valeurs particulières de la condition finale et de T .

La figure 4.3 montre la solution optimale en ε des mêmes problèmes, du point de vue de la norme sup de la différence entre les conditions finales théorique et estimée. La valeur optimale de ε (cherchée par dichotomie) est $2.2 \cdot 10^{-4}$, et l'erreur correspondante est légèrement inférieure à 2%. Cependant, comme cela a été évoqué précédemment, ε est trop petit et le problème Q.R. se rapproche de l'équation de la chaleur rétrograde, rendant les résolutions numériques particulièrement délicates, voire impossibles. On

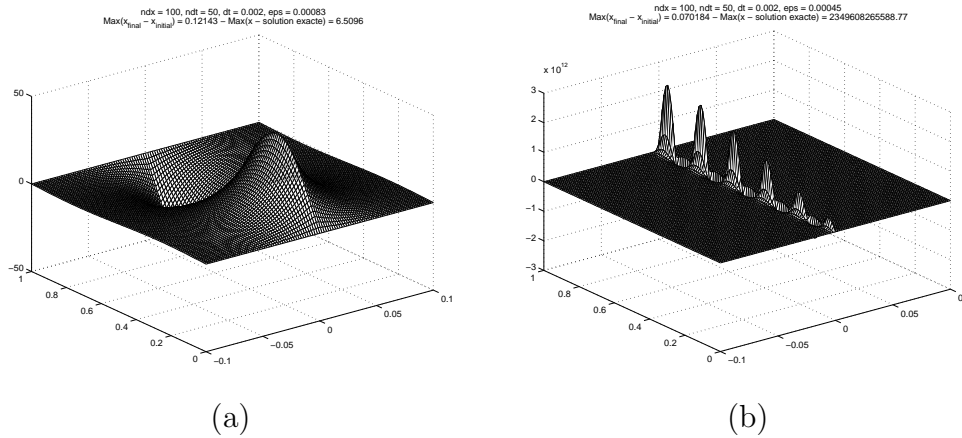


FIG. 4.4 – Solutions approchées des problèmes (4.7) et (4.5) avec $\varepsilon = 8.3 \cdot 10^{-4}$ (a) et $\varepsilon = 4.5 \cdot 10^{-4}$ (b), et $T = 0.1$

constate ainsi que la solution du problème Q.R. atteint des valeurs de l'ordre de 10^{12} et il est clair que la solution construite est aberrante, même si elle fournit l'erreur minimale sur la condition finale. Cependant, la condition finale obtenue est vraiment proche de χ , à quelques infimes oscillations près. Ceci s'explique vraisemblablement par le fait que l'équation de la chaleur a tendance à lisser les trajectoires. Cependant, nous voyons arriver les limites de la méthode Q.R.

Au vu de ces résultats, la longueur de l'intervalle de travail a été doublé, et désormais, T est égal à 0.1. Les figures 4.4-a et 4.4-b montrent les solutions numériques des problèmes Q.R. et direct pour $\varepsilon = 8.3 \cdot 10^{-4}$ et $\varepsilon = 4.5 \cdot 10^{-4}$ respectivement. Dans le premier cas, ε a été choisi pour que la solution Q.R. soit la plus proche de la solution exacte au sens de la norme infinie mesurée sur tout l'intervalle $[0, T]$, alors que dans le second cas, la valeur de ε a été choisie pour minimiser (toujours en norme infinie) l'écart uniquement entre les conditions finales. Dans le premier cas (a), l'erreur atteint 12% (contre 3 lorsque T était égal à 0.05), et dans le cas (b), l'erreur est de l'ordre de 7% (contre 2 précédemment), et une fois encore, la solution construite n'a aucun sens. Cette dernière remarque tend à montrer que la longueur de l'intervalle de travail est déjà trop importante. De même, les valeurs de ε croissent avec T et atteignent presque 10^{-2} , valeur presque trop grande pour considérer que l'on travaille encore sur l'équation de la chaleur.

Nous allons néanmoins tester plusieurs variantes de la quasi-réversibilité afin d'essayer d'obtenir de meilleurs résultats sur l'équation de la chaleur.

4.4 Variantes sur la quasi-réversibilité

La méthode Q.R. telle qu'elle a été définie par Lattès et Lions [24] a montré plusieurs limites du point de vue numérique qui risquent de la rendre inefficace lorsqu'on voudra l'appliquer à des problèmes d'océanographie par exemple. Nous allons tenter de la modifier pour essayer d'obtenir de meilleurs résultats.

La méthode Q.R. consiste à rajouter un terme comprenant l'opérateur au carré puisqu'il s'agit de remplacer l'opérateur $A(t)$ par $A(t) - \varepsilon A^*(t)A(t)$. Nous pouvons alors essayer de faire de même avec l'opérateur différentiel en temps $\frac{\partial}{\partial t}$, et donc rajouter un terme d'ordre deux (ou plus) en temps, c'est-à-dire remplacer l'opérateur de dérivation $\frac{\partial}{\partial t}$ par $\frac{\partial}{\partial t} - \varepsilon \frac{\partial^k}{\partial t^k}$, avec k variant de 2 (par analogie avec la méthode Q.R. originale) à 4 (pour faire apparaître une dérivée d'ordre 4 dans l'équation, comme dans la méthode Q.R. originale).

4.4.1 Q.R. d'ordre 2 en temps

L'équation considérée comme problème rétrograde est désormais la suivante :

$$\varepsilon \frac{\partial^2 u_\varepsilon}{\partial t^2} - \frac{\partial u_\varepsilon}{\partial t} = \frac{\partial^2 u_\varepsilon}{\partial x^2}. \quad (4.8)$$

On se réserve désormais le droit de prendre ε négatif puisqu'il n'est pas évident de savoir quel signe mettre devant le terme d'ordre 2 en temps. Si ε est choisi positif, l'équation (4.8) est du type équation des ondes, et il suffit de mettre les conditions aux bords suivantes pour obtenir l'existence et l'unicité de la solution :

$$\begin{aligned} u_\varepsilon(x, 0) &= \chi(x), & x &\in \Omega, \\ \frac{\partial u_\varepsilon}{\partial t}(x, 0) &= -\Delta \chi(x), & x &\in \Omega, \\ u_\varepsilon(x, t) &= 0, & x &\in \partial\Omega, \quad t \in]0, T[. \end{aligned}$$

Par contre, si ε est choisi négatif, l'équation (4.8) s'apparente à une équation de Laplace et il faudrait se donner une condition initiale, une condition finale et une condition sur les bords pour avoir un problème bien posé. Malheureusement, il est impossible de se donner numériquement une condition finale puisque c'est ce que l'on cherche à identifier. Nous allons donc imposer les mêmes conditions aux bords que pour ε positif dans un cadre strictement numérique.

Les figures 4.5-a et -b montrent les solutions optimales de ce dernier problème (puis, pour les temps positifs, les solutions de l'équation de la chaleur directe) pour des valeurs de ε choisies respectivement positives et négatives. L'intervalle de temps

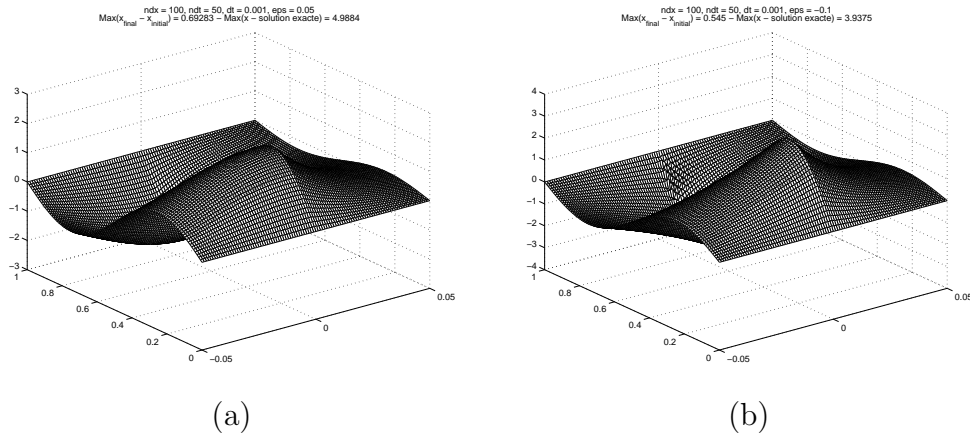


FIG. 4.5 – Solutions optimales du problème (4.8) pour des valeurs de ε respectivement positives (a) et négatives (b).

sur lequel ont été intégrées les solutions est de nouveau de longueur 0.05, car comme précédemment, lorsque celui-ci est choisi trop grand (0.1 ou plus), les résultats empirent assez vite.

Dans un cas comme dans l'autre, la solution construite est très mauvaise, avec une erreur relative de plus de 50%. Par contre, on peut noter une différence de courbure entre la figure 4.5-a et toutes les autres figures obtenues dans ce chapitre. Ceci intervient lorsque le paramètre ε est choisi parmi les réels strictement négatifs, et donc pour un problème de nature elliptique.

Cette méthode apparaissant comme nettement moins bonne que la méthode Q.R. originale, nous ne nous sommes pas attardés à faire de plus amples tests numériques.

4.4.2 Q.R. d'ordre supérieur en temps

L'équation que nous allons utiliser pour remonter le temps est maintenant la suivante :

$$\varepsilon \frac{\partial^k u_\varepsilon}{\partial t^k} - \frac{\partial u_\varepsilon}{\partial t} = \frac{\partial^2 u_\varepsilon}{\partial x^2} \quad (4.9)$$

avec k valant soit 3, soit 4, et ε pouvant de nouveau être positif ou négatif. L'intervalle de travail est encore de longueur 0.05. On se donne toujours une condition initiale $u_\varepsilon(x, 0) = \chi(x)$, on impose toujours la nullité de la solution sur $\partial\Omega$, et on se donne $\frac{\partial^i u_\varepsilon}{\partial t^i}(x, 0)$ pour i variant de 1 à $k - 1$, par analogie avec le paragraphe précédent.

Les solutions de cette équation sont représentées sur la figure 4.6, pour des valeurs de k choisies égales à 3 (a) et 4 (b) respectivement. Pour $k = 3$, la méthode ne donne absolument rien. En effet, on assiste à un changement de signe de la solution, puis à une tendance à diverger fortement. En partant après d'une telle condition initiale

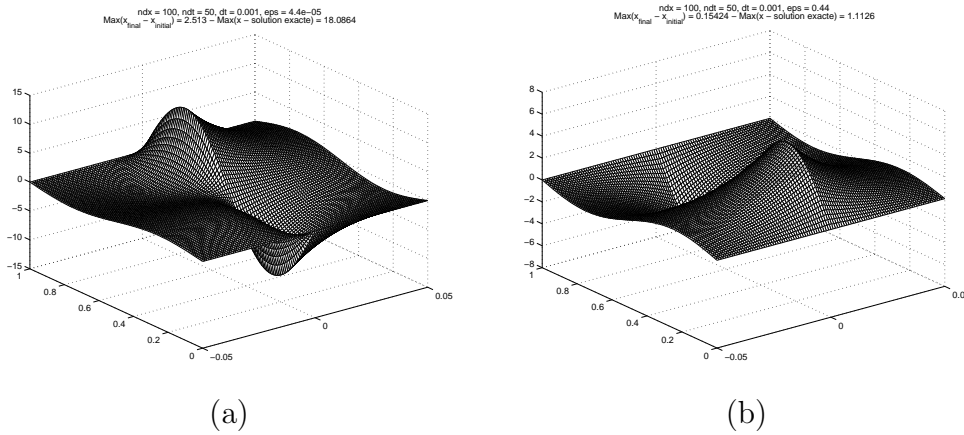


FIG. 4.6 – Solutions optimales du problème (4.9) avec $k = 3$ (a) et $k = 4$ (b).

pour le problème de la chaleur classique, non seulement la solution obtenue est trop grande en norme, mais elle est de signe opposé à ce à quoi on s'attend.

Par contre, lorsqu'on choisit k égal à 4, on retrouve une solution qui a la même allure générale qu'avec la méthode Q.R. classique. L'erreur minimale (mesurée en norme infinie entre χ et la condition finale reconstruite) est d'environ 15%, et donc bien supérieure à celle obtenue avec la méthode originale. De plus, ce minimum est atteint pour une valeur de ε égale à 0.44. Cette valeur est bien trop importante pour ne pas dénaturer l'équation de la chaleur.

D'autres tests ont été réalisés dans ce dernier cas ($k = 4$) mais aucun n'a permis d'obtenir des résultats plus concluants.

Il apparaît donc que la méthode Q.R. originale, bien que peu satisfaisante, reste la meilleure méthode testée ici avec l'équation de la chaleur. L'équation de la chaleur constitue le premier modèle sur lequel nous avons testé la quasi-réversibilité puisque c'est une équation d'évolution relativement simple, et dont l'instabilité rétrograde est fort bien connue. Néanmoins, cette équation est trop éloignée des systèmes habituellement utilisés en océanographie car, notamment, elle est linéaire. Il convient donc d'essayer la quasi-réversibilité sur un système différentiel non linéaire, à comportement chaotique.

4.5 Autres systèmes différentiels testés

Des tests ont tout d'abord été réalisés sur le système de Lorenz, un système non linéaire chaotique relativement simple. Nous présentons rapidement les équations du

système de Lorenz (il sera étudié plus en détail dans la suite de ce chapitre) :

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = \rho x - y - xz, \\ \frac{dz}{dt} = \beta z + xy, \end{cases} \quad (4.10)$$

avec $\sigma = 10$, $\rho = 28$ et $\beta = \frac{8}{3}$. En notant $X = (x; y; z)$ le vecteur position de \mathbb{R}^3 et $F(X) = (\sigma(y - x); \rho x - y - xz; \beta z + xy)$ le vecteur vitesse, le but du problème est de résoudre le système différentiel *rétrograde*

$$\begin{aligned} \frac{dX}{dt} &= F(X), \quad t \in [0, T], \\ X(T) &= X_T, \end{aligned} \quad (4.11)$$

c'est-à-dire d'identifier la condition initiale ayant conduit à X_T à l'instant final. En remplaçant t par $T - t$, on se ramène au problème d'évolution suivant :

$$-\frac{dX}{dt} = F(X), \quad t \in [0, T], \quad (4.12)$$

avec une condition initiale $X(0) = X_0$ (qui est en fait la condition finale X_T du problème rétrograde).

Directement, cela est numériquement impossible. En effet, la solution diverge avant le temps $t = 1$ alors que les trajectoires du système direct n'auraient pas oscillé plus d'une fois dans le même temps.

Les équations Q.R. considérées sont les suivantes :

$$-\frac{dX}{dt} = F(X) + \varepsilon F(F(X))$$

et

$$(4.13)$$

$$\varepsilon \frac{d^2 X}{dt^2} - \frac{dX}{dt} = F(X),$$

avec ε de signe a priori quelconque. La première équation est l'application directe de la quasi-réversibilité au système de Lorenz (en ajoutant l'opérateur F au carré) et la seconde équation correspond à un opérateur de quasi-réversibilité en temps.

Malheureusement, aucune de ces deux équations n'a donné de résultats numériques intéressants : la trajectoire diverge toujours aussi rapidement, et il n'est pas envisageable de travailler sur des intervalles de longueur inférieure à la période moyenne

(de l'ordre de 1) des oscillations des trajectoires directes. Les autres variantes de la méthode Q.R. présentées précédemment ont également été testées sur le système de Lorenz, toujours sans succès.

Des tests sommaires ont également été réalisés sur un modèle océanique quasi-géostrophique barotrope, toujours sans succès.

4.6 Conclusions sur la quasi-réversibilité

Au vu des résultats numériques, il apparaît que la quasi-réversibilité permet d'intégrer de façon rétrograde des équations telles que l'équation de la chaleur, mais sur des intervalles de temps assez courts et dans des cas très particuliers de condition initiale (en prenant une fonction qui n'est plus de classe C^∞ , la méthode ne permet plus de reconstruire correctement la solution). De plus, dès qu'on essaie d'appliquer la méthode Q.R. sur des systèmes d'équations non linéaires et chaotiques tels que le système de Lorenz ou un océan quasi-géostrophique barotrope, les résultats numériques n'ont plus rien donné d'intéressant. Ceci tend à montrer que la quasi-réversibilité n'a pas beaucoup d'intérêt en océanographie, et ne remplacera pas des méthodes telles que le nudging (direct ou rétrograde) dans la recherche rapide d'une condition initiale à partir d'une condition finale.

4.7 Introduction sur le nudging

La technique dite du nudging a été introduite par Anthes en 1974 [3]. Elle consiste à rajouter un terme de rappel aux observations dont on dispose dans les équations différentielles régissant un système à caractère chaotique afin de s'assurer que les trajectoires ne s'éloignent pas trop des observations. Sans un tel terme de rappel, la moindre perturbation de l'état initial peut conduire très rapidement à une trajectoire erronée, s'éloignant inexorablement des observations. Cette méthode permet ainsi de laisser évoluer librement la trajectoire lorsqu'aucune observation du système n'est disponible, et de la contrôler lorsque des observations sont disponibles afin qu'elle ne diverge pas.

Nous allons tout d'abord détailler la méthode du nudging avant d'étudier le nudging rétrograde, qui n'est autre qu'une application du nudging direct à des problèmes rétrogrades, réputés numériquement instables. Nous verrons ensuite des applications numériques de ces deux méthodes à deux systèmes différentiels non linéaires, tout d'abord le système de Lorenz, puis un modèle océanique quasi-géostrophique barotrope.

4.8 Nudging direct

4.8.1 Principe de la méthode

Le principe du nudging est le suivant : considérons un système dynamique dont l'état à l'instant t est défini par un vecteur $X(t)$ de \mathbb{R}^n , et régi par une équation différentielle du type

$$\frac{dX}{dt} = F(X). \quad (4.14)$$

On suppose que l'on dispose d'un jeu d'observations d'une partie du système, effectuées sur une période d'assimilation $[0, T]$ à intervalles réguliers de temps h : $X_{obs}(t_0), X_{obs}(t_1), \dots, X_{obs}(t_N)$, avec $t_0 = 0$, $t_i = t_0 + ih$ et $t_N = t_0 + Nh = T$.

Ces observations sont des vecteurs de \mathbb{R}^p avec $p \leq n$. En effet, généralement, les observations ne concernent qu'une partie du système. Par exemple, il est assez difficile d'obtenir des observations océaniques en profondeur et elles sont en quantité négligeable par rapport aux observations de surface. Il faut donc un opérateur C de projection de l'espace des vecteurs d'état \mathbb{R}^n dans celui des vecteurs d'observation \mathbb{R}^p . Nous avons également besoin d'une matrice $K \in \mathcal{M}_{np}(\mathbb{R})$ (appelée plus tard *matrice de nudging*) permettant de passer d'un vecteur d'observations X_{obs} à un vecteur d'état X .

Les observations dont on dispose étant réalisées par des stations-sondes ou des satellites par exemple dans le cas de la météorologie, elles ne sont pas parfaites. Nous appellerons dans la suite erreur d'observation la différence entre les observations et les états réels du système $\varepsilon(t_i) = X_{obs}(t_i) - CX(t_i)$.

Le nudging consiste alors à chercher une solution X de l'équation (4.14) qui soit la plus proche possible de l'échantillonnage X_{obs} . Pour cela, on rajoute un terme de rappel aux observations aux instants t_i :

$$\frac{dX}{dt} = F(X) + \sum_{i=0}^N K(X_{obs}(t_i) - CX(t_i)) \cdot \delta(t - t_i) \quad (4.15)$$

Nous verrons dans la suite que le choix des coefficients de K est très important puisque si K est choisie trop petite, l'équation (4.15) sera pratiquement identique à l'équation (4.14) et la présence d'observations disponibles ne sera pratiquement pas exploitée. À l'inverse, si les coefficients de la matrice de nudging sont choisis trop grands, la solution ainsi fabriquée ne sera plus représentative des équations primitives du modèle. De plus, les différents coefficients de K permettent de répartir à toutes les coordonnées des vecteurs d'état l'information disponible uniquement sur certaines composantes. Cela permet par exemple dans le cas d'un océan de propager l'information souvent disponible uniquement en surface à tous les niveaux de profondeur.

L'avantage de cette méthode par rapport à des méthodes plus classiques d'assimilation de données est sa simplicité et son faible coût de mise en œuvre lors de tests

numériques. En effet, à part le choix de K , certes important, la méthode ne consiste qu'à intégrer l'équation (4.15) pour obtenir une trajectoire passant relativement près de toutes les observations et à partir de laquelle des prévisions futures peuvent être obtenues en intégrant simplement le système initial (4.14). Nous verrons dans les parties concernant les applications numériques la qualité des prévisions que l'on pourra en déduire.

4.8.2 Justification théorique dans un cadre linéaire

Nous allons vérifier sur un exemple linéaire que le terme de rappel introduit dans l'équation conduit bien à stabiliser la solution autour des observations. Considérons un problème simplifié

$$\frac{dX}{dt} = FX + K(X_{obs} - X) \quad (4.16)$$

où F est linéaire et où l'opérateur d'observation C est égal à l'identité. Notons $X(0) = X_0$ la condition initiale. Nous supposons enfin que $K - F$ est inversible, afin de simplifier le problème. Alors la solution du problème est

$$X(t) = (I - e^{-t(K-F)}) (K - F)^{-1} K X_{obs} + e^{-t(K-F)} X_0$$

et il est clair que si on fait tendre t vers $+\infty$ et que la matrice $K - F$ est positive, la solution $X(t)$ tendra vers la solution du problème stationnaire $(K - F)^{-1} K X_{obs}$. Si on choisit la matrice K suffisamment grande (i.e. avec des valeurs propres suffisamment grandes), on assure la positivité de $K - F$ et, de plus, la solution sera alors proche de X_{obs} au voisinage de $+\infty$.

On remarquera également que si on choisit la matrice K *trop petite* (de sorte que toutes les valeurs propres de la matrice $K - F$ soient négatives), alors la solution s'éloigne exponentiellement en temps de X_{obs} .

4.8.3 Remarques sur la méthode

L'écriture du problème sous la forme (4.16) permet de faire apparaître un lien assez clair avec les méthodes séquentielles reposant sur le filtre de Kalman. En effet, si la matrice de nudging K est convenablement choisie, le nudging ainsi utilisé se ramène à un filtrage de Kalman, et en conséquence, pour un problème linéaire, le nudging optimal est équivalent au filtre de Kalman.

En prenant l'opérateur d'observation H égal à l'identité et en écrivant un schéma temporel aux différences finies pour (4.16), on peut écrire le problème sous la forme

$$\frac{X_{n+1}^a - X_n^a}{t_{n+1} - t_n} = F_n X_n^a + K_{n+1} (X_{obs_{n+1}} - X_{n+1}^f)$$

où X_n^a est l'état analysé à l'instant t_n , K_n la matrice de nudging à l'instant t_n , et X_n^f l'ébauche de l'état du système à l'instant t_n .

En résolvant d'abord l'équation (4.14) du temps t_n au temps t_{n+1} en partant de X_n^a pour construire l'ébauche X_{n+1}^f , on retrouve la formule (2.9) du filtre de Kalman. En remplaçant X_{n+1}^f par $X_n^a + (t_{n+1} - t_n)F_n X_n^a$, la formule donnant X_{n+1}^a est alors exactement l'équation (2.11) donnant le nouvel état analysé dans la méthode du filtre de Kalman. L'équivalence des deux méthodes est alors vérifiée en choisissant à chaque instant comme matrice de nudging la matrice de gain du filtre de Kalman.

4.9 Nudging rétrograde

Le nudging peut être utilisé d'une autre manière. Il peut en effet servir à stabiliser l'intégration rétrograde du système (4.14). Dans l'optique de la reconstitution de l'état initial, une méthode simple et rapide consiste à intégrer rétrogradement l'équation différentielle régissant le système en partant d'une condition finale mal connue en espérant récupérer une meilleure estimation de l'état initial. En faisant un changement de variable en temps $t' = T - t$ dans l'équation (4.14), le système différentiel devient :

$$-\frac{dX}{dt'} = F(X) \quad (4.17)$$

avec une condition finale en t et donc une condition initiale en t' . On peut réécrire cette équation sous la forme

$$\frac{dX}{dt} = -F(X) \quad (4.18)$$

en remplaçant t' par t , et en ajoutant une condition initiale $X(0)$.

Néanmoins, un problème se pose, celui de réussir à intégrer rétrogradement le système différentiel (4.14) avec une condition finale ou, de façon équivalente, à intégrer directement le système (4.18) avec une condition initiale. En effet, l'instabilité numérique des systèmes rétrogrades est telle qu'il s'avère souvent impossible de les intégrer, même sur de courtes périodes de temps. L'idée du nudging rétrograde consiste justement à stabiliser l'équation rétrograde avec un terme de rappel aux observations (le même terme que pour le nudging direct) :

$$\frac{dX}{dt} = -F(X) + \sum_{i=0}^N K(X_{obs}(t_i) - CX(t_i)) \cdot \delta(t - t_i) \quad (4.19)$$

Non seulement l'intégration rétrograde sera numériquement possible, mais en plus la trajectoire obtenue sera relativement proche de toutes les observations, ce qui tend à supposer que l'état initial ainsi calculé constituera une bonne estimation de l'état initial cherché et permettra de faire de bonnes prévisions.

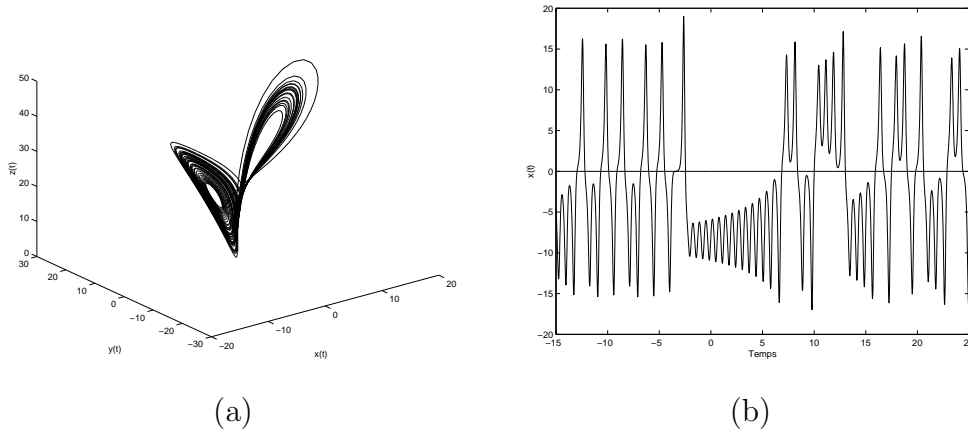


FIG. 4.7 – Évolution dans le temps du vecteur d'état (x, y, z) du système de Lorenz (a), et variations de la variable x en fonction du temps pour les mêmes conditions initiales

4.10 Application au système de Lorenz

Nous allons maintenant appliquer ces différentes techniques de nudging à un système non linéaire chaotique à trois dimensions, le système de Lorenz.

4.10.1 Présentation des équations du système de Lorenz

Le système de Lorenz est un jeu de trois équations différentielles non linéaires couplées faisant intervenir les trois variables $(x, y, z) \in \mathbb{R}^3$:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x), \\ \frac{dy}{dt} = \rho x - y - xz, \\ \frac{dz}{dt} = \beta z + xy, \end{cases} \quad (4.20)$$

où σ , ρ et β sont des constantes réelles positives. Dans la suite, ces constantes pour valeurs : $\sigma = 10$, $\rho = 28$ et $\beta = \frac{3}{8}$.

La figure 4.7-a montre l'évolution du vecteur $(x(t), y(t), z(t))$ sur l'intervalle de temps $[-15, 25]$, obtenue en intégrant le système de Lorenz à partir de la condition initiale $(-4.62, -6.61, 17.94)$ au temps $t = -15$. La figure 4.7-b correspond à l'évolution de la variable $x(t)$ en fonction du temps à partir des mêmes conditions initiales. Les deux régimes du système de Lorenz correspondent à des oscillations autour de deux points fixes instables du système, symétriques par rapport à l'axe $x = y = 0$. Les

oscillations autour de chacun des points fixes, pendant lesquelles la variable x garde un signe constant, sont clairement visibles sur la figure 4.7, et ont une période de l'ordre de 1. Leur amplitude augmente progressivement, et après quelques oscillations, une transition a lieu vers l'autre point fixe. Les transitions correspondent à un changement de signe de x .

En notant $X = (x; y; z)$ le vecteur position et $F(X) = (\sigma(y - x); \rho x - y - xz; \beta z + xy)$ le vecteur vitesse, le système de Lorenz s'écrit exactement sous la forme de l'équation (4.14).

4.10.2 Nudging rétrograde appliqué au système de Lorenz

Le nudging rétrograde appliqué au système de Lorenz s'écrit donc sous la forme de l'équation (4.19). Dans le cas présent, l'espace d'observation est exactement l'espace \mathbb{R}^3 , c'est-à-dire qu'on observe les trois composantes du vecteur d'état. Le système étant naturellement couplé, il s'avère inutile de coupler les observations à l'aide de la matrice de nudging K . Cette dernière se présentera donc sous la forme d'une matrice carrée diagonale à 3 dimensions :

$$K = \begin{pmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{pmatrix}$$

et l'opérateur de projection C est la matrice identité. Le système rétrograde s'écrit donc sous la forme :

$$\begin{cases} \frac{dx}{dt} = -\sigma(y - x) + K_1 \sum_{i=0}^N (x_{obs}(t_i) - x) \cdot \delta(t - t_i), \\ \frac{dy}{dt} = -\rho x + y + xz + K_2 \sum_{i=0}^N (y_{obs}(t_i) - y) \cdot \delta(t - t_i), \\ \frac{dz}{dt} = -\beta z - xy + K_3 \sum_{i=0}^N (z_{obs}(t_i) - z) \cdot \delta(t - t_i) \end{cases} \quad (4.21)$$

4.10.3 Résultats numériques du nudging rétrograde

La figure 4.8 montre la trajectoire de chacune des trois coordonnées du point courant en fonction du temps. x et y sont en bas et z en haut. Pour un temps t compris entre 0 et 1, on voit la trajectoire du système direct issue du point $(-4.62; -6.61; 17.94)$, les croix représentant les instants où on effectue des observations (ici, $N = 10$). On relève à ces instants les coordonnées exactes du point.

Pour un temps t compris entre 1 et 2, on voit la solution du problème rétrograde avec nudging (4.21) utilisant comme condition initiale l'observation du système au

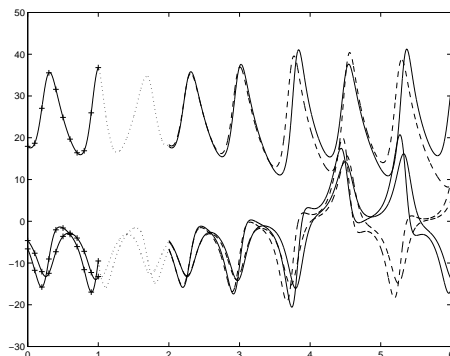


FIG. 4.8 – Nudging rétrograde sur le système de Lorenz

temps T (ici, 1). L'impression de symétrie par rapport au temps $t = 1$ vient du fait qu'entre les instants $t = 1$ et $t = 2$, on remonte le temps en partant de la condition finale obtenue au temps $t = 1$ à partir du problème direct. Les constantes de nudging K_1 , K_2 et K_3 ont été choisies de façon optimale, ni trop faible pour éviter à la solution de diverger, ni trop forte pour ne pas dénaturer le problème. Les intégrations des différentes équations étant suffisamment rapides, ces constantes ont été cherchées par dichotomie de sorte à fournir les meilleurs résultats sur plusieurs expériences dont les conditions initiales ont été choisies aléatoirement.

À partir du temps $t = 3$ sont représentées d'une part en pointillés la solution du problème direct (sans nudging) avec comme condition initiale la solution finale du problème rétrograde précédent, et d'autre part, en trait plein, la solution exacte du problème, en utilisant la condition initiale $(-4.62; -6.61; 17.94)$.

On constate alors que pour des temps compris entre 2 et presque 5, la trajectoire issue de la solution du problème rétrograde avec nudging reste proche de la trajectoire de référence, et vers le temps $t = 5$, la trajectoire change de point fixe attracteur et va osciller autour d'un autre point fixe. Cette méthode est donc satisfaisante puisque, pour une durée (environ 3) supérieure à celle sur laquelle on a travaillé (1) en rétrograde, la trajectoire approchée que l'on calcule reste très proche de la trajectoire réelle.

Ces résultats nous permettent d'affiner le modèle en rajoutant des erreurs dans les observations. Les figures 4.9-a et -b sont analogues à la figure 4.8 dans le cas où les observations relevées contiennent des erreurs de mesure, de l'ordre de 5% pour la figure 4.9-a et de 20% pour la figure 4.9-b. De plus, nous avons représenté à partir du temps $t = 3$ (équivalent au temps $t = 1$, fin de la période d'assimilation et début de la période de prévisions) en pointillés fins la trajectoire issue de la dernière observation réalisée au temps $t = 1$.

On constate que la figure 4.9-a ressemble beaucoup à la figure 4.8, ce qui s'explique par le faible taux d'erreurs d'observation (néanmoins suffisant pour voir les croix

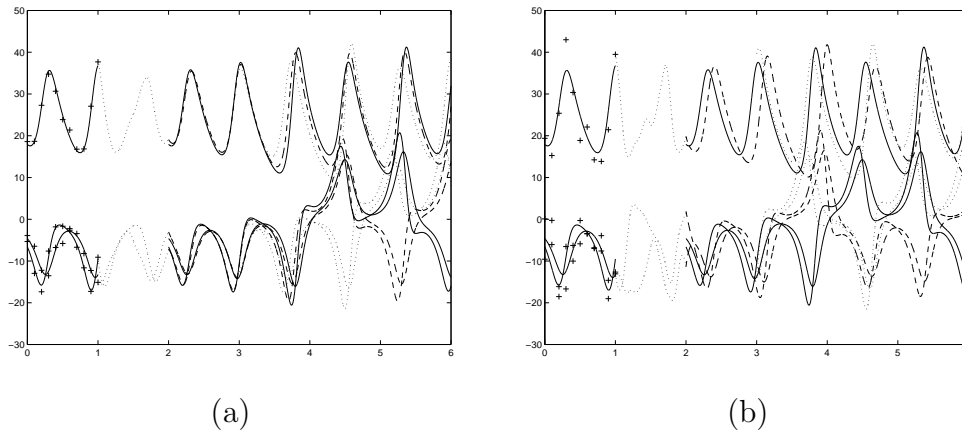


FIG. 4.9 – Nudging rétrograde appliqué au système de Lorenz, avec des observations contenant des erreurs de l'ordre de 5% (a) et 20% (b).

représentant les observations réalisées en dehors de la trajectoire réelle). De plus, on constate qu'à partir de l'instant $t = 4$, la trajectoire issue de la dernière observation (en pointillés fins) s'éloigne de la trajectoire réelle (en trait plein) avant la trajectoire reconstruite par le nudging rétrograde. Ceci tend à démontrer une certaine efficacité et utilité de la méthode d'assimilation.

Par contre, lorsque les erreurs sont plus importantes (figure 4.9-b), la trajectoire prédite s'éloigne assez vite de la trajectoire réelle (après à peine plus d'une oscillation pour x et y), tout comme la trajectoire issue de la dernière observation (en pointillés fins).

Le nudging rétrograde semble donc relativement bien adapté à des problèmes hautement non linéaires, mais nous avons mis en avant les limites de cette méthode dans le cas du système de Lorenz : il faut que les observations soient suffisamment fiables, et ne pas travailler sur de trop grandes périodes de temps.

4.11 Application à un modèle quasi-géostrophique barotrope

Nous allons maintenant appliquer le nudging direct et le nudging rétrograde à un modèle océanique, le modèle quasi-géostrophique barotrope ([21], [40]).

4.11.1 Équations du modèle

Équations de base Les deux équations mécaniques sur lesquelles tout le modèle est construit expriment la conservation de la quantité de mouvement (équations

de Navier-Stokes) et la conservation de la masse (équation de continuité). La dynamique quasi-géostrophique néglige quasiment tous les aspects thermodynamiques pour ne conserver que les équations mécaniques. Nous allons voir comment, à partir des équations de base, on obtient les équations quasi-géostrophiques au prix de plusieurs hypothèses simplificatrices. Le détail complet de ce cheminement est expliqué dans Pedlosky [40].

En coordonnées sphériques (r, θ, λ) (θ représentant la latitude et λ la longitude) dans un repère ayant son origine au centre de la terre, elles s'écrivent :

$$\left\{ \begin{array}{l} \frac{Du}{Dt} + \frac{uw}{r} - \frac{uw}{r} \tan \theta - 2\Omega v \sin \theta + 2\Omega w \cos \theta = -\frac{1}{\rho r \cos \theta} \frac{\partial p}{\partial \lambda} + \frac{\mathcal{F}_\lambda}{\rho} \\ \frac{Dv}{Dt} + \frac{uv}{r} + \frac{u^2}{r} \tan \theta + 2\Omega u \sin \theta = -\frac{1}{\rho r} \frac{\partial p}{\partial \theta} + \frac{\mathcal{F}_\theta}{\rho} \\ \frac{Dw}{Dt} - \frac{u^2}{r} - \frac{v^2}{r} - 2\Omega u \cos \theta = -\frac{1}{\rho} \frac{\partial p}{\partial r} - g + \frac{\mathcal{F}_r}{\rho} \end{array} \right. \quad (4.22)$$

et

$$\frac{D\rho}{Dt} + \rho \left[\frac{\partial \rho}{\partial r} + \frac{2w}{r} + \frac{1}{r \cos \theta} \frac{\partial(v \cos \theta)}{\partial \theta} + \frac{1}{r \cos \theta} \frac{\partial u}{\partial \lambda} \right] = 0. \quad (4.23)$$

Il y a 5 inconnues (u, v, w, p et ρ) donc il faut rajouter une cinquième équation pour compléter le système. Comme la quasi-géostrophie néglige la thermodynamique, les échanges de chaleur sont donc considérés comme nuls (adiabacité du système). Nous allons supposer de plus que la densité ρ du fluide vérifie :

$$\frac{D\rho}{Dt} = 0. \quad (4.24)$$

Dans ces équations,

- u, v et w sont les composantes de la vitesse dans le repère sphérique ;
- ρ désigne la densité et p la pression du fluide ;
- g représente l'accélération de la pesanteur ;
- \mathcal{F} représente les forces de frictions internes au fluide ;
- enfin, $\frac{D}{Dt}$ est la dérivée lagrangienne suivant le mouvement :

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \frac{u}{r \cos \theta} \frac{\partial}{\partial \lambda} + \frac{v}{r} \frac{\partial}{\partial \theta} + w \frac{\partial}{\partial r}$$

Adimensionnalisation Nous allons faire quelques hypothèses en vue de simplifier ces équations, afin d'obtenir l'approximation quasi-géostrophique, mais tout d'abord, il faut adimensionner les variables. On note respectivement L, D et U les échelles caractéristiques de distance horizontale, de distance verticale, et de vitesse horizontale. L'échelle de temps caractéristique est alors $\frac{L}{U}$ et celle de vitesse verticale $\frac{DU}{L}$.

Considérons les nombres sans dimension suivants :

- $\delta = \frac{D}{L}$ est le rapport d'aspect, rapport entre la profondeur et la largeur de l'océan ;
- $\varepsilon = \frac{U}{f_0 L}$ est le nombre de Rossby, f_0 étant la valeur du paramètre de Coriolis $f = 2\Omega \sin \theta$ à la latitude centrale θ_0 . Le nombre de Rossby, rapport entre $\frac{1}{f_0}$ et $\frac{L}{U}$, mesure donc l'importance relative des effets d'inertie et de rotation terrestre. Aux latitudes moyennes, ε est petit, ce qui signifie que les temps caractéristiques de l'écoulement du fluide sont beaucoup plus grands que l'échelle de temps de la rotation terrestre ;
- $F = \frac{f_0^2 L^2}{gD} = \left(\frac{L}{R_{ext}}\right)^2$, R_{ext} étant appelé le rayon externe de déformation. F mesure donc l'échelle à laquelle l'effet de la gravité (tendant à aplanir la surface libre) est contrebalancé par la force de Coriolis (tendant à la déformer).

Simplification des équations Les hypothèses émises sur le modèle sont les suivantes :

1. L'approximation géostrophique : $\varepsilon \ll 1$ et $\delta \ll 1$, ce qui revient à supposer que l'effet d'inertie est faible devant l'effet de la rotation de la terre, et que l'océan est assimilé à une couche mince à l'échelle planétaire.
2. $\frac{D}{L} = \mathcal{O}(\varepsilon)$. Cette approximation est réaliste, surtout aux moyennes échelles.
3. $\frac{L}{r_0} = \mathcal{O}(\varepsilon)$, r_0 étant le rayon de la terre. Cette approximation n'est pas valable aux grandes échelles, mais elle permet de linéariser le paramètre de Coriolis en fonction de la latitude. Il s'agit de l'approximation du β -plan : $f = f_0 + \beta_0 y$ avec $\beta_0 = \frac{2\Omega}{r_0} \cos \theta_0$.

Développements par rapport à ε Une fois ces approximations faites, il ne reste plus qu'à développer les équations par rapport à ε ([40]). Le développement à l'ordre 0 conduit à des relations entre la force de Coriolis et le gradient de pression horizontal, exprimant l'équilibre géostrophique.

Nous allons considérer dans la suite un modèle simplifié d'océan :

- le bassin est supposé fermé et de profondeur constante,
- la circulation est induite par la seule tension du vent à la surface de la mer,
- la dissipation est principalement due aux frottements sur le fond du bassin, mais se fait aussi par frottement latéral,
- et enfin, le fluide est supposé homogène (ρ constant) : c'est la modélisation *barotrope*.

En posant $\beta = \frac{\beta_0 L^2}{U}$ le paramètre du β -plan, $\delta_f = \frac{\eta}{\beta_0 L}$ et $\delta_i = \frac{1}{\beta^{\frac{1}{2}}}$, le développement à l'ordre 1 donne alors :

$$\left\{ \begin{array}{l} \frac{\partial \xi}{\partial t} + \delta_i^2 J(\psi, \xi) + \frac{\partial \psi}{\partial x} = \text{rot}_z(V) - \delta_f \xi \quad \text{sur } \Omega \\ \Delta \psi = \xi \quad \text{sur } \Omega \\ \xi = \psi = 0 \quad \text{sur } \partial\Omega \\ \xi(t=0) = \xi_0 \end{array} \right. \quad (4.25)$$

où ψ est la fonction de courant géostrophique, directement liée à la pression, V l'intensité du vent, et η le coefficient de frottement de fond. J est l'opérateur jacobien : $J(a, b) = \frac{\partial a}{\partial x} \frac{\partial b}{\partial y} - \frac{\partial a}{\partial y} \frac{\partial b}{\partial x}$. Enfin, on note $\xi = \Delta \psi$ la vorticité. Les deux paramètres sans dimension δ_f et δ_i caractérisent la largeur du courant de bord ouest du bassin. L'épaisseur de ce courant sur le bord est contrôlé par la dissipation par frottement de fond (δ_f) et par l'inertie (δ_i). Le terme $\frac{\partial \psi}{\partial x}$ correspond à la force de Coriolis. Enfin, la vitesse horizontale $\begin{pmatrix} u \\ v \end{pmatrix}$ se déduit de la fonction de courant :

$$u = -\frac{\partial \psi}{\partial y} \quad \text{et} \quad v = \frac{\partial \psi}{\partial x}. \quad (4.26)$$

Conditions aux bords Nous allons maintenant discuter du choix des conditions aux bords du système (4.25). Soit $\Gamma = \partial\Omega$ la frontière du bassin. Nous imposerons sur la vitesse horizontale $(u; v)$ une condition de glissement à la paroi :

$$\begin{pmatrix} u \\ v \end{pmatrix} \cdot n = 0 \quad \text{sur } \Gamma, \quad (4.27)$$

où n est la normale extérieure au domaine. Si τ est la tangente à la frontière, la condition précédente traduite en terme de fonction de courant devient $\frac{\partial \psi}{\partial \tau} = 0$ sur Γ . On en déduit donc que ψ est constant sur Γ . La frontière coïncide donc avec une ligne de courant. De plus, ψ étant défini à une constante près, on peut imposer sur le bord $\psi = 0$.

En reprenant la condition au bord (4.27), on a $\frac{\partial((u; v) \cdot n)}{\partial \tau} = 0$. En développant cette dernière équation, on obtient :

$$\left(\frac{\partial(u; v)}{\partial n} + \frac{1}{R}(u; v) \right) \cdot \tau + \xi = 0$$

en appelant R le rayon de courbure de la frontière au point considéré. Pour un domaine carré ou rectangulaire, R est infini, donc son inverse est nul, et choisir $\xi = 0$ le long de Γ revient à considérer que la vitesse de l'écoulement le long des frontières n'est pas perturbée normalement à celles-ci. C'est la condition que l'on s'impose.

Les conditions aux bords choisies sont donc :

$$\begin{cases} \psi = 0 & \text{sur } \Gamma, \\ \Delta\psi = \xi = 0 & \text{sur } \Gamma. \end{cases} \quad (4.28)$$

Énergie cinétique Le principal moteur des circulations océaniques est, dans notre modèle, le vent. L'énergie induite par la tension du vent en surface est redistribuée dans le bassin par les termes $\frac{\partial\psi}{\partial x}$ et $J(\psi, \xi)$, et dissipée par le frottement de fond. L'énergie cinétique est donnée par :

$$E = \int_{\Omega} \nabla\psi \cdot \nabla\psi = - \int_{\Omega} \psi\xi. \quad (4.29)$$

On peut donc calculer la variation de l'énergie en fonction du temps :

$$\frac{dE}{dt} = - \int_{\Omega} \frac{\partial\psi}{\partial t} \xi - \int_{\Omega} \frac{\partial\xi}{\partial t} \psi.$$

La première intégrale vaut alors

$$- \int_{\Omega} \frac{\partial\psi}{\partial t} \xi = - \int_{\Omega} \frac{\partial\psi}{\partial t} \Delta\psi = \int_{\Omega} \nabla\psi \cdot \nabla \frac{\partial\psi}{\partial t} - \int_{\Gamma} \frac{\partial\psi}{\partial n} \frac{\partial\psi}{\partial t}$$

en utilisant la formule de Green. Comme ψ est nul sur le bord à tout instant, on obtient :

$$- \int_{\Omega} \frac{\partial\psi}{\partial t} \xi = \frac{1}{2} \frac{\partial}{\partial t} \int_{\Omega} \|\nabla\psi\|^2 = \frac{1}{2} \frac{dE}{dt}.$$

On en déduit alors, en utilisant (4.25), que

$$\frac{dE}{dt} = -2 \left(\int_{\Omega} \psi \cdot \text{rot}_z(V) - \delta_f \int_{\Omega} \psi\xi - \delta_i^2 \int_{\Omega} \psi J(\psi, \xi) - \int_{\Omega} \psi \frac{\partial\psi}{\partial x} \right).$$

Dès que ψ est nul ou ξ est constant sur le bord (ce qui est le cas ici), on a $\int_{\Omega} \psi J(\psi, \xi) = 0$. En intégrant par parties, on obtient $\int_{\Omega} \psi \frac{\partial\psi}{\partial x} = 0$. On en déduit donc :

$$\frac{dE}{dt} = -2 \left(\int_{\Omega} \psi \cdot \text{rot}_z(V) + \delta_f \int_{\Omega} \|\nabla\psi\|^2 \right). \quad (4.30)$$

En régime stationnaire, l'équilibre se fait donc entre l'énergie fournie par la tension du vent en surface et celle dissipée par frottement au fond du bassin.

Les termes advectif $\int_{\Omega} \psi J(\psi, \xi)$ et de Coriolis $\int_{\Omega} \psi \frac{\partial \psi}{\partial x}$ n'interviennent pas dans la balance d'énergie cinétique. Ils redistribuent dans le bassin l'énergie fournie par le vent.

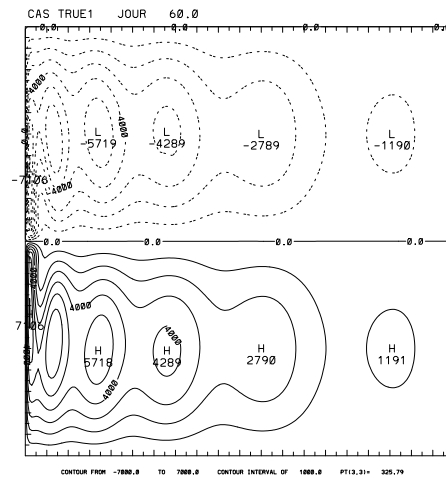
Paramètres du modèle L'océan considéré est carré, de 4000 km de côté. Sa profondeur est de 5 kilomètres. L'océan est discrétisé suivant une grille carrée de 200×200 éléments. Le pas de temps utilisé pour les intégrations numériques est égal à 5400 secondes, soit une heure et demie.

4.11.2 Utilisation du nudging pour faire des prévisions

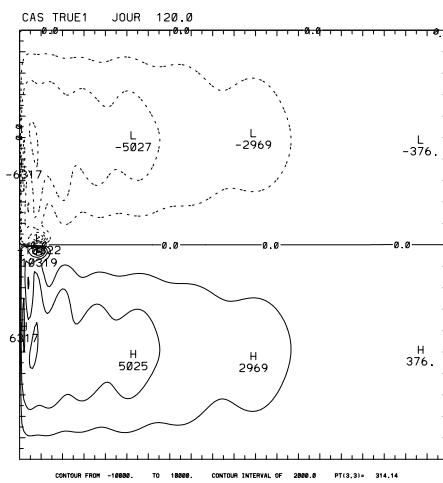
Le modèle quasi-géostrophique barotrope a été numériquement intégré sur une période de temps de quelques mois en partant d'une condition initiale constante (champ initial constant). À partir de l'état du système obtenu, dont on dira qu'il correspond au temps $-T$, et pendant une période de deux mois (on prend T égal à deux mois), le modèle a été intégré et des observations de tout le système ont été relevées tous les deux jours. Le champ obtenu à la fin de la période de deux mois sera considéré comme correspondant au temps 0 (que l'on pourra assimiler au temps présent). Puis, en prenant comme condition initiale ce dernier champ, le modèle a encore été intégré sur une période de temps de deux mois, pour fournir des prévisions du système jusqu'au temps T (prévisions sur 2 mois). Les champs représentés sur les figures 4.10-a, -b et -c sont les lignes de niveau de la fonction de courant, respectivement aux instants $-T$, 0 et T .

Prévisions directes Les observations faites entre les temps $-T$ et 0 ont ensuite été biaisées suivant une loi normale centrée, afin d'obtenir un champ perturbé d'environ 5%. La figure 4.11-a montre l'observation du système au temps 0 ainsi obtenue, à comparer avec la figure 4.10-b montrant le champ de référence au même instant. Le modèle quasi-géostrophique barotrope a été intégré à partir de cette dernière observation sur une période de temps de deux mois, afin d'obtenir une première série de prévisions. La figure 4.11-b montre le champ obtenu au temps T , que l'on peut comparer avec le champ de référence de la figure 4.10-c.

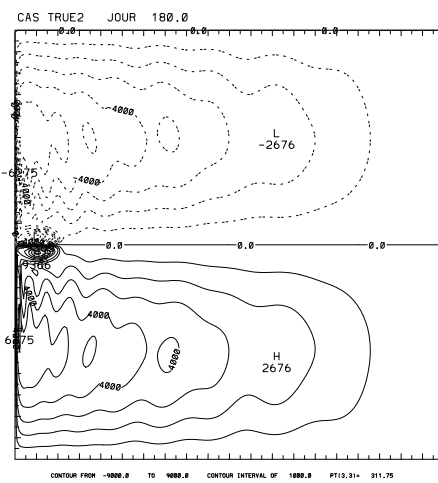
On constate que la différence entre la trajectoire exacte et la trajectoire issue de la dernière observation (biaisée) évolue peu, et le champ obtenu au temps T reste assez éloigné du champ exact au même instant. Il apparaît donc très peu intéressant de faire des prévisions directement à partir des dernières observations, du fait de l'erreur d'observation assez importante (ici de l'ordre de 5%). Néanmoins, c'est un



(a)



(b)



(c)

FIG. 4.10 – Champs de référence du modèle quasi-géostrophique respectivement au temps $-T$ (a), 0 (b) et T (c), avec T égal à deux mois.

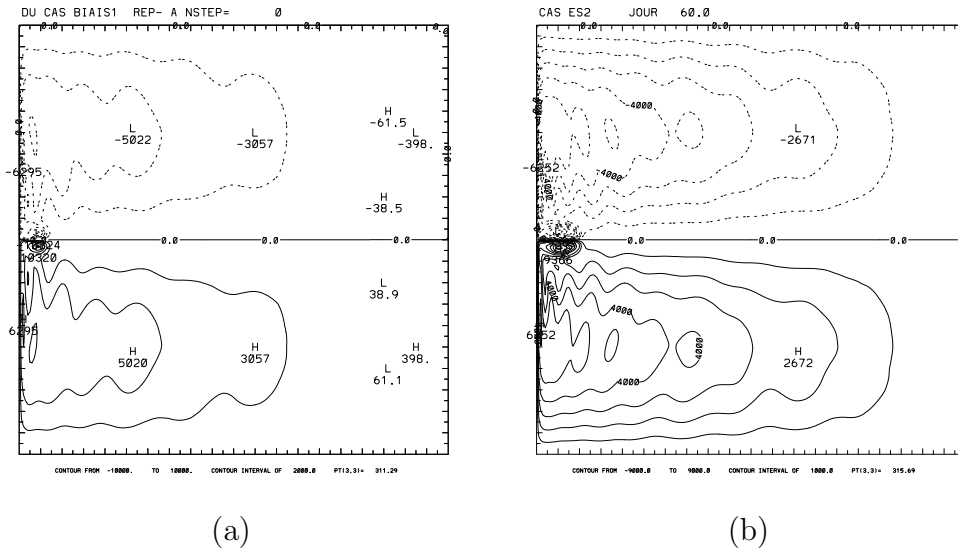


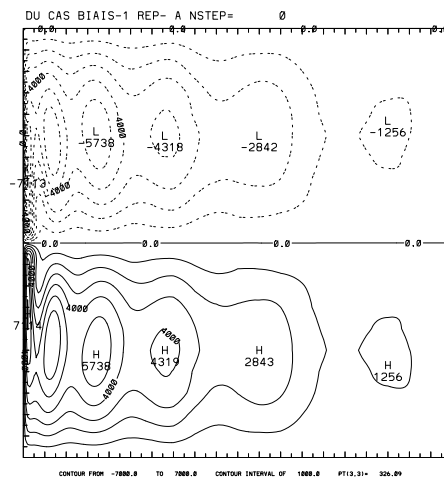
FIG. 4.11 – Champ biaisé correspondant au temps 0 (a) et prévision obtenue au temps T en partant de ce dernier (b).

moyen rapide d’obtenir, en première approximation, des prévisions afin d’étudier globalement le comportement futur du système.

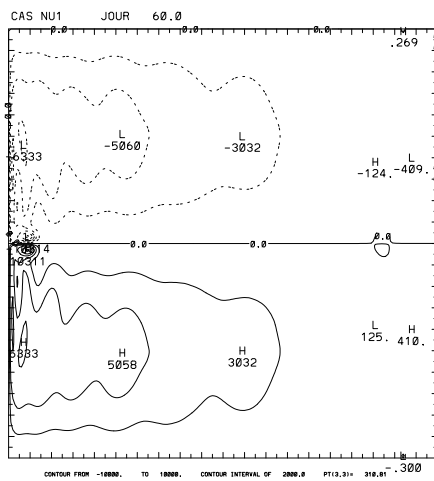
Dans la suite, pour chaque série d’expériences numériques, la matrice de nudging a été choisie égale à un multiple de l’identité (afin de décorréler les observations entre elles), la constante ayant été cherchée par dichotomie afin de produire les meilleurs résultats possibles dans quelques cas tests indépendants des expériences suivantes.

Nudging direct Nous avons ensuite testé la méthode du nudging direct en partant, au temps $-T$, de la première observation dont on dispose (observation biaisée du système à l’instant $-T$), que l’on peut visualiser sur la figure 4.12-a, sur une période de deux mois et en utilisant toutes les observations dont on dispose sur cette période. Le champ obtenu au temps 0 est représenté sur la figure 4.12-b. En prenant ce dernier comme condition initiale, le modèle direct (sans nudging) a ensuite été intégré pour obtenir une seconde série de prévisions du système (toujours entre les instants 0 et T). Le champ obtenu au temps T est représenté sur la figure 4.12-c.

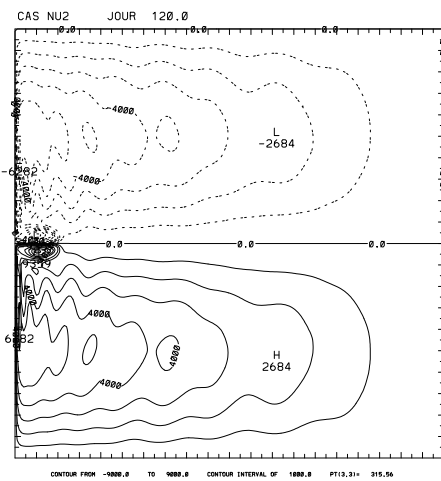
La figure 4.12-a montre à quel point le champ peut être perturbé par les erreurs d’observation. Néanmoins, le nudging direct permet de se rapprocher de la trajectoire exacte, et à l’instant 0, la trajectoire semble déjà plus proche de la trajectoire de référence que celle issue directement des observations (figure 4.11). La prévision obtenue au temps T est donc meilleure que dans le cas précédent, mais est encore peu intéressante (comme nous le verrons plus loin dans un récapitulatif).



(a)



(b)



(c)

FIG. 4.12 – Observation biaisée du système à l’instant $-T$ (a), champ obtenu à l’aide de la méthode du nudging direct au temps 0 (b), et prévision déduite correspondant au temps T .

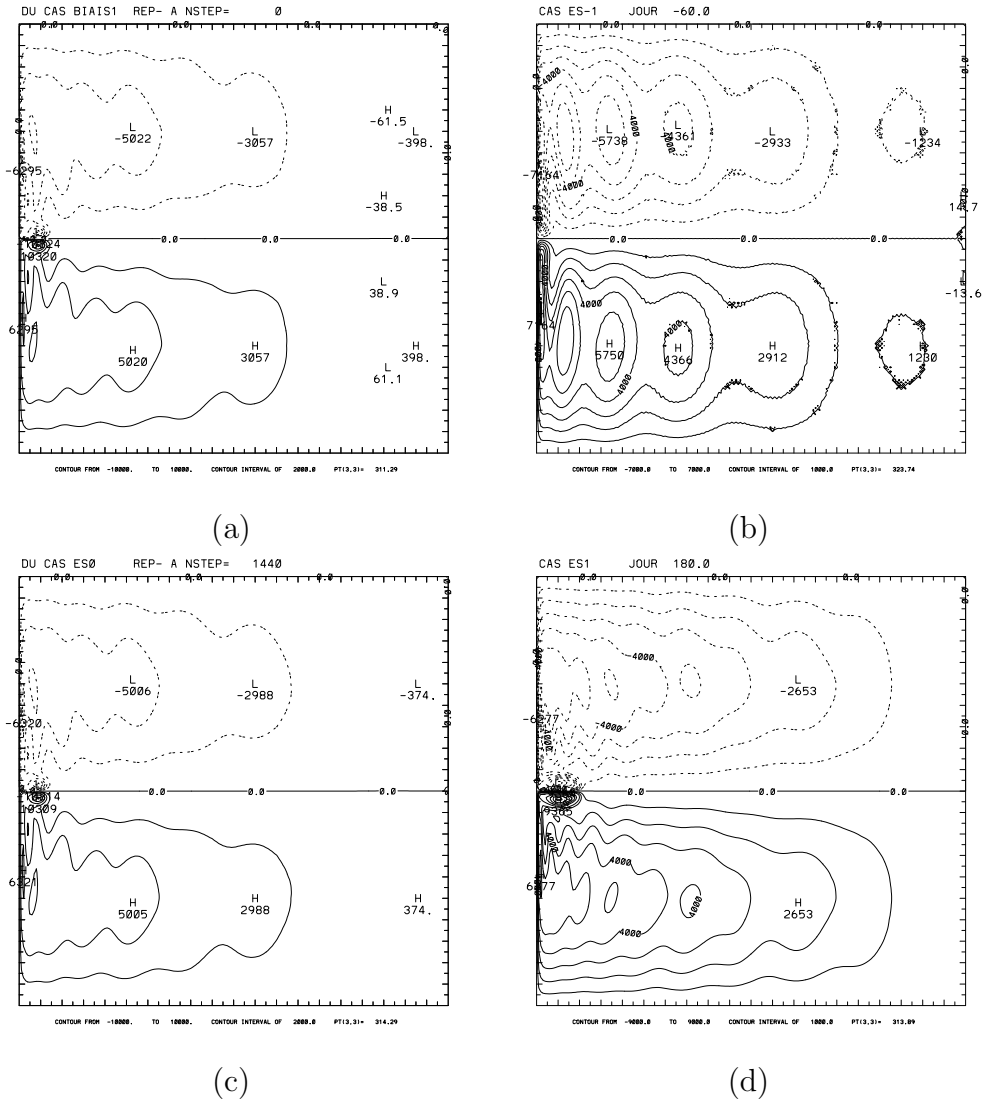


FIG. 4.13 – Observation biaisée correspondant au temps 0 (a), champ obtenu au temps $-T$ à partir de cette dernière en utilisant la méthode du nudging rétrograde (b), estimation du système au temps 0 déduite en intégrant directement le modèle quasi-géostrophique barotrope (c), et prévision correspondante obtenue au temps T (d).

Nudging rétrograde Nous avons ensuite testé le nudging rétrograde sur ce modèle. En partant de la dernière observation disponible (celle au temps 0), représentée sur la figure 4.13-a, le modèle a été intégré de façon rétrograde, avec un terme de nudging, jusqu'au temps $-T$. Le champ obtenu est représenté sur la figure 4.13-b. Le modèle quasi-géostrophique est ensuite intégré directement, sans nudging, en partant du dernier champ obtenu, pendant une durée de deux mois, et on obtient une nouvelle estimation de l'état du système au temps 0, représentée sur la figure 4.13-c, et à partir de laquelle on peut faire des prévisions sur une période de deux mois. La prévision obtenue au bout de ces deux mois (temps T) est représentée sur la figure 4.13-d.

On constate sur la figure 4.13-b que le champ obtenu après intégration rétrograde du système avec nudging ressemble assez au champ exact au temps $-T$ (figure 4.10-a), mais des irrégularités apparaissent sur les lignes de niveau. Lorsque l'intégration rétrograde (même avec nudging) est poussée sur plusieurs mois supplémentaires, ces irrégularités augmentent et finissent par brouiller complètement le champ. Une certaine limite à cette méthode apparaît donc. En effet, bien qu'apparemment numériquement stable, le problème rétrograde a tendance à diverger lentement (et de toute façon beaucoup plus lentement que sans terme de nudging).

Une manière de relisser le champ, et donc de stabiliser la trajectoire rétrograde, est d'augmenter la constante de nudging, afin de se rapprocher des observations qui ne sont pas aussi brouillées, mais là encore, des difficultés apparaissent : la trajectoire se met à osciller trop brutalement autour des observations et finit par diverger. Une solution consisterait à faire dépendre du temps cette constante (avec par exemple une exponentielle décroissante pour ramener le champ vers les observations au début, puis le laisser évoluer plus librement par la suite).

Lorsque le système est intégré de façon directe, les perturbations visibles sur la figure 4.13-b disparaissent, et le modèle a tendance à lisser le champ. Une fois l'instant 0 atteint, on constate que la trajectoire s'est considérablement rapprochée de la trajectoire exacte, et les prévisions en sont d'autant meilleures.

L'intégration directe du système pour passer de l'instant $-T$ à 0 a été réalisée sans nudging, mais il semble judicieux d'essayer de mettre un terme de rappel aux observations afin d'essayer de se rapprocher encore un peu plus de la trajectoire de référence. Cela n'a en pratique aucun effet et n'améliore les prévisions que de façon insensible.

Amélioration Pour tenter d'améliorer encore les prévisions, nous avons appliqué deux fois de suite la méthode du nudging rétrograde. La figure 4.14-a montre l'estimation de l'état présent (au temps 0) du système obtenue à l'aide du nudging rétrograde, ce qui n'est autre qu'un rappel de la figure 4.13-c. Le nudging rétrograde à partir de ce champ permet de trouver une estimation correspondant au temps $-T$, représentée sur la figure 4.14-b, à partir de laquelle on obtient une nouvelle estimation de l'état

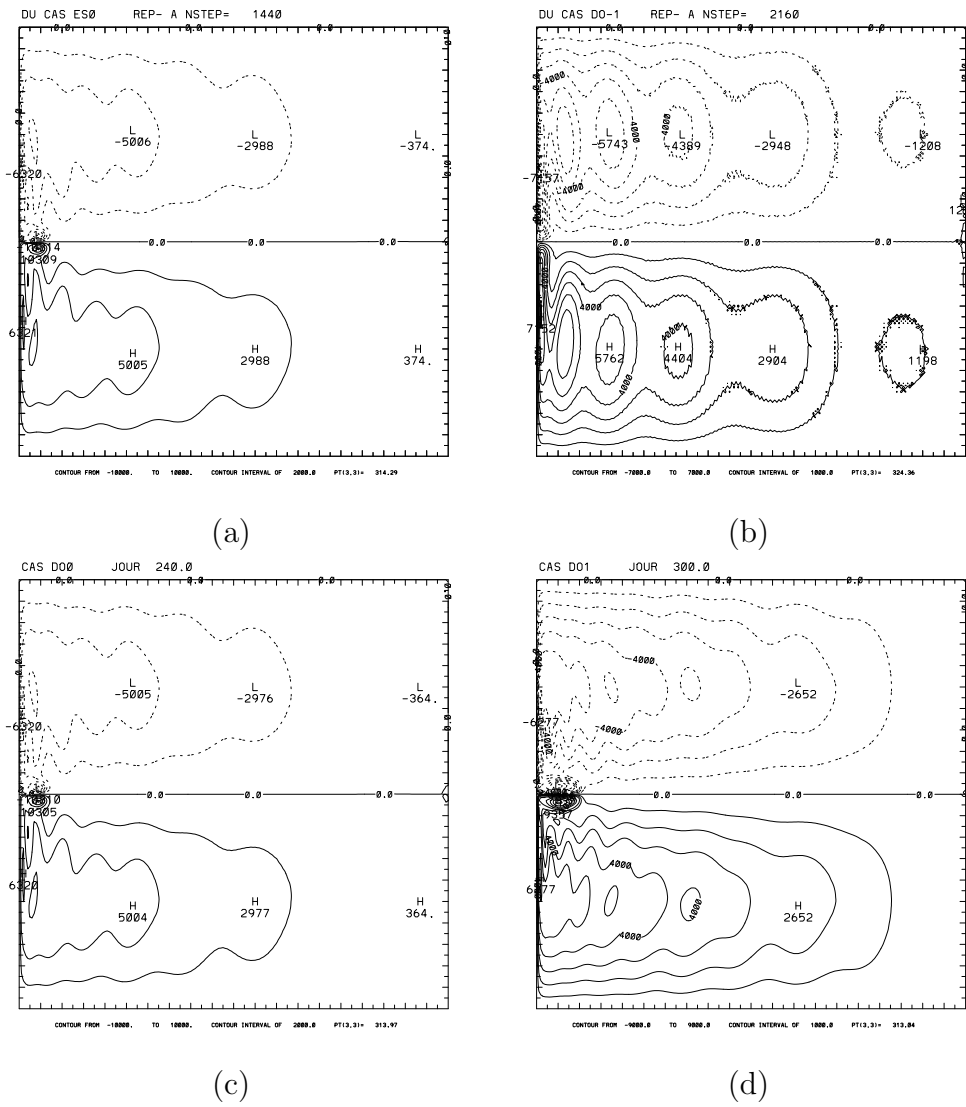


FIG. 4.14 – Etat du système au temps 0 obtenu par la première application du nudging rétrograde (a), champ obtenu au temps $-T$ en réappliquant le nudging rétrograde (b), estimation de l'état présent du système (c), et prévision correspondante au temps T (d).

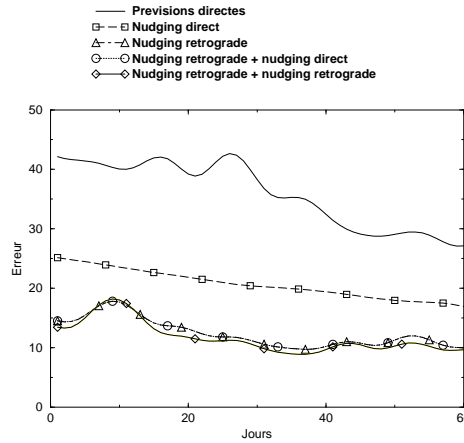


FIG. 4.15 – Différence en norme L^2 entre la trajectoire réelle et les trajectoires obtenues grâce aux différentes méthodes de nudging entre les instants 0 et T .

du système à l'instant 0 (figure 4.14-c), et la prévision correspondante, au temps T (figure 4.14-d).

Là encore, le champ obtenu après intégration rétrograde apparaît comme très peu lisse (encore moins que lorsque l'on avait appliqué une seule fois le nudging rétrograde en figure 4.13-b, ce qui montre que l'intégration directe n'avait pas totalement effacé ces irrégularités), mais une fois encore, après intégration directe (avec ou sans nudging), le champ redevient à peu près lisse et proche de la trajectoire réelle. Les prévisions obtenues sont alors très fiables, et légèrement meilleures que lorsque la méthode du nudging rétrograde n'est appliquée qu'une seule fois.

Conclusions sur les prévisions La figure 4.15 montre, pour chaque simulation numérique testée, la différence relative en norme L^2 (ramenée à la norme L^2 de la trajectoire exacte) entre la trajectoire exacte et les prévisions calculées, exprimée en pour mille, en fonction du temps (entre les instants 0 et T , donc entre 0 et 60 jours). On voit sur cette figure :

- en trait plein sans symboles, les prévisions obtenues directement à partir de la dernière observation disponible (figure 4.11) ;
- avec des carrés, les prévisions obtenues à partir du résultat de la méthode du nudging direct appliquée sur la période $[-T, 0]$ (figure 4.12) ;
- avec des triangles, les prévisions obtenues à partir de la méthode du nudging rétrograde, puis intégration directe sans nudging du système pour revenir à l'instant 0 (figure 4.13) ;
- avec des ronds, les prévisions obtenues également après intégration rétrograde du système avec nudging, puis intégration directe du système avec cette fois-ci un terme de nudging pour revenir au temps 0 ;

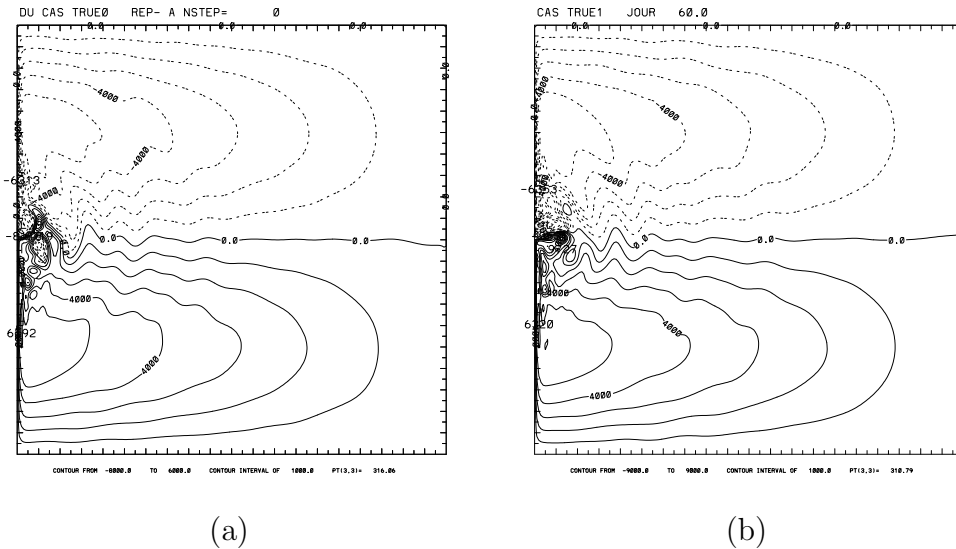


FIG. 4.16 – États exacts du système aux instants $-T$ (a) et 0 (b).

- avec des losanges, les prévisions obtenues après deux utilisations successives des méthodes du nudging rétrograde puis nudging direct (figure 4.14).

On voit assez clairement le gain apporté par le nudging rétrograde par rapport aux prévisions directement issues des dernières observations, et aussi par rapport au nudging direct (même en partant d'un champ pas trop éloigné de la trajectoire réelle) : la différence entre la trajectoire exacte et la trajectoire simulée a été divisée par 3 ou 4 grâce au nudging rétrograde. Le gain obtenu en appliquant deux fois de suite le nudging rétrograde est relativement faible, et il n'est donc pas intéressant de renouveler plus de deux fois de suite le nudging rétrograde, la simulation numérique prenant d'autant plus de temps que la méthode est réappliquée.

4.11.3 Reconstitution numérique de l'état initial

Nous allons maintenant étudier l'utilité du nudging rétrograde dans la reconstitution de l'état initial. En assimilation variationnelle de données, le principal problème est la reconstitution de l'état initial de la période d'assimilation. Celle-ci est souvent partielle, et a actuellement un coût de mise en œuvre très élevé. Or, comme nous l'avons vu dans les paragraphes précédents, le nudging rétrograde permet d'estimer l'état correspondant au début de la période d'assimilation à partir d'une estimation de l'état final. Nous allons étudier numériquement la qualité de cette reconstitution.

Le modèle quasi-géostrophique barotrope a de nouveau été intégré sur quelques années (cette fois-ci un peu plus longtemps afin d'avoir plus d'instabilités dans les champs pour voir comment se comportent les méthodes de nudging dans ce genre de

cas) afin d'obtenir un champ que l'on pourra qualifier d'état du système au temps $-T$, avec T toujours égal à deux mois. Sur l'intervalle de temps $[-T, 0]$ pendant lequel le modèle a été numériquement intégré, des observations complètes (toujours biaisées de la même façon que précédemment) du système ont été relevées tous les deux jours. On obtient à la fin de l'intégration le champ correspondant à l'instant 0. La figure 4.16 montre les champs obtenus aux instants $-T$ et 0.

Généralement, l'estimation de l'état initial est mauvaise, et pour en tenir compte, nous avons pris comme estimation de départ de l'état initial, l'état du système à l'instant $-2T$, donc l'état réel du système, mais deux mois auparavant. Afin d'appliquer le nudging rétrograde, nous devons commencer par intégrer le modèle dans le sens direct sur une période de deux mois, donc jusqu'au temps 0. Alors, en utilisant les observations disponibles tout au long de la période d'assimilation, le nudging rétrograde permettra d'obtenir une nouvelle estimation de l'état initial. Il y a deux solutions pour obtenir une estimation de l'état au temps 0 à partir de celle au temps $-T$, qui est a priori très inexacte : intégrer le modèle direct, avec ou sans nudging.

Avec nudging La figure 4.17-a montre le champ correspondant à l'estimation de l'état initial. Le nudging direct entre les instants $-T$ et 0 permet de trouver une estimation de l'état final de la période d'assimilation, représentée en figure 4.17-b. Puis, le nudging rétrograde permet de revenir à l'instant $-T$, et on obtient une nouvelle estimation de l'état initial, représentée sur la figure 4.17-c.

Le champ de départ (figure 4.17-a) est assez éloigné de l'état initial réel (figure 4.16-a), et on voit que le nudging direct (avec une constante de nudging assez élevée) permet de se rapprocher beaucoup de la trajectoire exacte. Ensuite, le nudging rétrograde permet de ne pas trop s'en éloigner (sans nudging, l'intégration rétrograde du système conduit à une estimation de l'état initial beaucoup plus fautive que celle de départ). On a donc amélioré l'estimation de l'état initial, même si le champ obtenu présente à nouveau des irrégularités qu'il conviendrait de faire disparaître en lissant le champ afin d'obtenir une estimation encore meilleure de l'état initial.

Sans nudging Les figures 4.18-a, -b et -c montrent les états du système de façon analogue aux figures 4.17, sauf que la reconstitution de l'état final (apparaissant sur la figure 4.18-b) à partir de l'estimation de l'état initial s'est faite en intégrant de façon directe le modèle, sans nudging. Le retour à l'instant $-T$ a eu lieu de la même manière qu'avant, avec la méthode du nudging rétrograde.

Lors de l'intégration directe, sans nudging, la trajectoire est restée éloignée de la trajectoire exacte, mais le nudging rétrograde a ensuite permis de s'en rapprocher (en divisant l'écart par deux environ). Au temps $-T$, le champ obtenu est alors très peu différent de celui obtenu dans le cas précédent, où l'intégration directe avait eu lieu avec un terme de nudging. Il apparaît donc que le nudging rétrograde semble peu

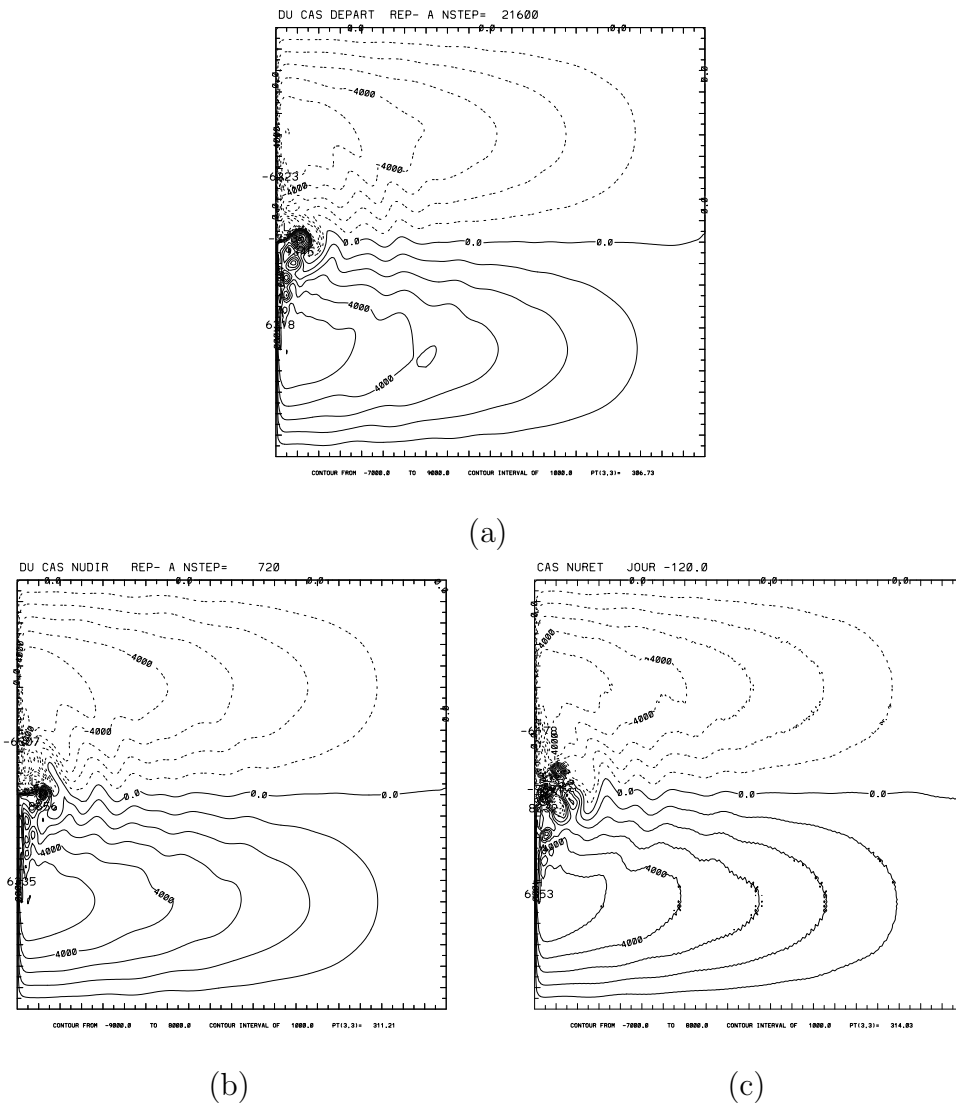


FIG. 4.17 – Estimation de l'état initial de la période d'assimilation (a), estimation de l'état final, obtenue avec du nudging direct (b), et nouvelle estimation de l'état initial en appliquant la technique du nudging rétrograde (c).

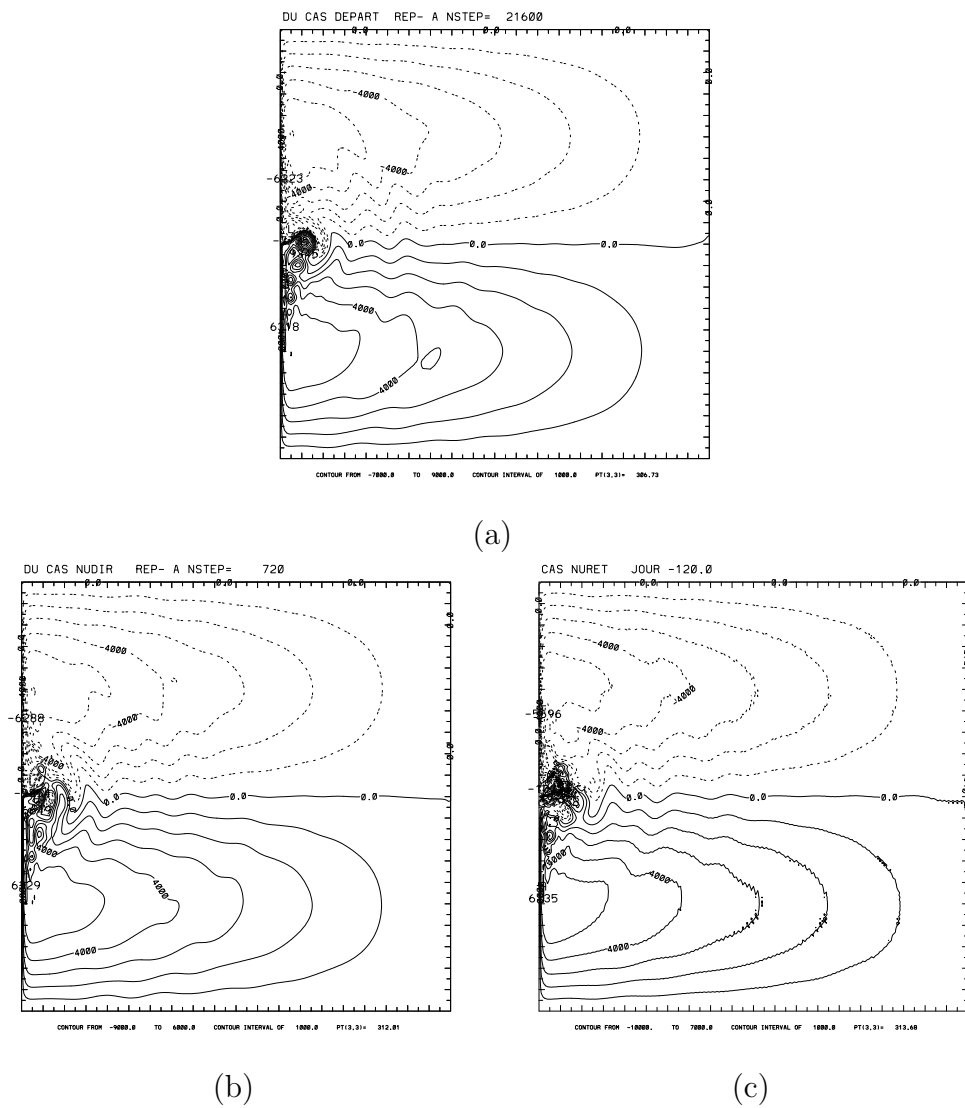


FIG. 4.18 – Estimation de l'état initial du système (a), estimation de l'état final par simple intégration du modèle (b), et nouvelle estimation de l'état initial par nudging rétrograde (c).

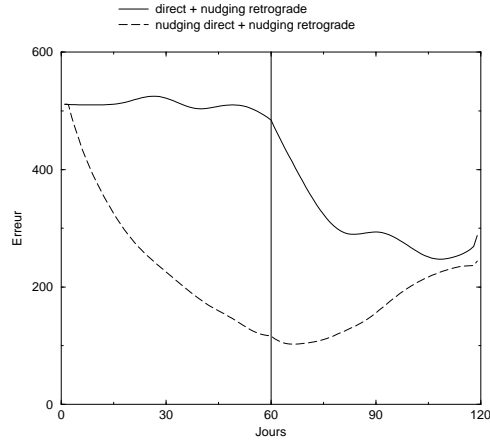


FIG. 4.19 – Différence en norme L^2 entre la trajectoire réelle et la trajectoire simulée lors de l'intégration du système direct (pour les temps compris entre 0 et 60 jours), avec ou sans nudging, et lors de l'intégration rétrograde (pour des temps compris entre 60 et 120 jours).

affecté par la présence ou non d'un terme de nudging lors des intégrations directes du système.

Conclusions sur la reconstitution de l'état initial La figure 4.19 montre, pour des temps compris entre 0 et 60 jours, la différence en norme L^2 entre la trajectoire exacte et la trajectoire simulée obtenue à partir de l'estimation de l'état initial grâce à une intégration directe du modèle quasi-géostrophique barotrope, avec (pointillés) ou sans (trait plein) nudging. Pour des temps compris entre 60 et 120 jours, on peut voir la différence entre la trajectoire exacte et la trajectoire calculée à partir de l'estimation de l'état final obtenu précédemment et avec la technique du nudging rétrograde.

On constate que le nudging direct permet de diviser par 5 la différence entre la trajectoire simulée et la trajectoire exacte, mais lorsque l'on est déjà proche de la trajectoire exacte, le nudging rétrograde n'apporte pratiquement rien, si ce n'est le contrôle de la trajectoire (pour l'empêcher d'exploser à l'infini), alors que lorsque l'on est éloigné de la trajectoire exacte, le nudging rétrograde permet de s'en rapprocher, avant de finalement, au bout d'un certain temps, s'en rééloigner. On peut également remarquer que, comme dans le paragraphe précédent portant sur les prévisions, il conviendrait d'augmenter la constante de nudging tout en s'assurant que le filtrage opéré par le terme de nudging lisse suffisamment les champs, afin de pouvoir obtenir éventuellement de meilleurs résultats.

4.12 Conclusions sur le nudging

La méthode du nudging rétrograde introduite dans ce chapitre offre une alternative aux problèmes actuels en assimilation variationnelle de données, à savoir notamment le très grand coût de mise en œuvre numérique. En effet, le nudging rétrograde ne nécessite que très peu d'intégrations (directes ou rétrogrades) du système d'équations considéré, contrairement aux méthodes d'assimilation variationnelle basées sur l'adjoint de premier ordre, qui nécessitent quant à elles plusieurs intégrations pour chaque étape d'un algorithme de descente visant à rechercher le minimum d'une fonctionnelle.

Les différentes expériences numériques menées sur le modèle quasi-géostrophique barotrope ont montré, sous réserve d'ajuster de façon optimale les constantes de nudging, que le nudging rétrograde permettait d'obtenir de très bonnes prévisions à deux mois, même à partir d'observations biaisées à hauteur de 5%. De plus, nous avons vu qu'il était possible de réduire par au moins deux l'erreur commise sur l'estimation de l'état initial, et les méthodes d'assimilation variationnelle de données reposant sur le principe de l'adjoint au premier ordre fournissent alors plus rapidement de meilleurs résultats.

Comme nous l'avons suggéré, il semble y avoir plusieurs améliorations possibles de la méthode du nudging rétrograde : constante de nudging variable au cours du temps, et donc possibilité de la choisir optimale à chaque instant et non plus globalement (voir [56]); lissage du champ après intégration rétrograde afin d'effacer les petites perturbations.

Chapitre 5

Dualité

5.1 Modèle physique

Nous allons travailler dans cette section sur un modèle océanique quasi-géostrophique barocline.

5.1.1 Quasi-géostrophie

Tout comme le modèle quasi-géostrophique barotrope étudié au chapitre 4, ce modèle résulte des équations de Navier-Stokes après avoir fait quelques simplifications et émis quelques hypothèses (voir la section (4.5) pour de plus amples détails). Nous devons tout d'abord supposer que l'effet de la rotation de la Terre, provenant de la force de Coriolis, est supérieur à l'effet d'inertie. Cela se mesure par le rapport entre le temps caractéristique de la rotation terrestre et le temps inertiel, rapport appelé nombre de Rossby. Il faut donc supposer que ce paramètre est petit devant 1. La quasi-géostrophie suppose également que l'océan est petit à l'échelle de la Terre, toujours dans un rapport de l'ordre du nombre de Rossby. Il s'agit de l'approximation du β -plan. Il faut enfin émettre l'hypothèse que l'océan est assimilable à une couche mince de la Terre, c'est-à-dire que la profondeur du bassin est petite en comparaison de sa largeur. Le modèle quasi-géostrophique barocline est souvent utilisé pour modéliser l'océan Atlantique Nord. Pour cet océan, toutes les hypothèses émises précédemment ne sont pas vérifiées. Néanmoins, il a été prouvé que ces approximations conduisent à un modèle qui reproduit assez bien les circulations océaniques aux latitudes intermédiaires, comme le Gulf Stream ou le jet.

Les effets thermodynamiques sont négligés. En effet, le comportement de la plupart des fluides géophysiques à grande échelle repose sur l'équilibre géostrophique entre l'effet rotationnel et le gradient de pression. On suppose enfin que le forçage de l'océan ne se fait que par le vent en surface, et la dissipation de l'énergie s'effectue par friction au bord et au fond.

5.1.2 Équations du modèle

Le modèle quasi-géostrophique barocline à n couches est un système de n équations couplées résultant de la loi de conservation du potentiel de vortacité. Les équations peuvent s'écrire de la façon suivante :

- pour la couche de surface ($k = 1$) :

$$\frac{D_1(\theta_1(\Psi) + f)}{Dt} - \beta\Delta^2\Psi_1 = F_1 \quad \text{dans } \Omega \times]0, T[; \quad (5.1)$$

- pour les couches intermédiaires ($2 \leq k \leq n - 1$) :

$$\frac{D_k(\theta_k(\Psi) + f)}{Dt} - \beta\Delta^2\Psi_k = 0 \quad \text{dans } \Omega \times]0, T[; \quad (5.2)$$

- pour la couche du fond ($k = n$) :

$$\frac{D_n(\theta_n(\Psi) + f)}{Dt} + \alpha\Delta\Psi_n - \beta\Delta^2\Psi_n = 0 \quad \text{dans } \Omega \times]0, T[. \quad (5.3)$$

Les notations sont les suivantes :

- n désigne le nombre de couches du modèle.
- Le domaine Ω est le bassin.
- Ψ_k représente la fonction de courant dans la couche k .

- On note Ψ le vecteur $\begin{pmatrix} \Psi_1 \\ \vdots \\ \Psi_n \end{pmatrix}$.

- θ_k est la somme de la vortacité dynamique ($\Delta\Psi_k$) et de la vortacité thermique ($-(W\Psi)_k$) pour la couche k :

$$\theta_k(\Psi) = \Delta\Psi_k - (W\Psi)_k$$

avec

$$-(W\Psi)_k = \frac{f_0^2 \rho}{h_k g} \left(\frac{\Psi_{k+1} - \Psi_k}{\rho_{k+1} - \rho_k} - \frac{\Psi_k - \Psi_{k-1}}{\rho_k - \rho_{k-1}} \right).$$

Il faut noter que la vortacité thermique peut être assimilée à un laplacien vertical, pondéré par les densités du fluide dans les différentes couches.

- f désigne la force de Coriolis, f_0 étant sa valeur à la latitude de référence de l'océan.

Dans l'approximation du β -plan, la force de Coriolis varie linéairement en fonction de la latitude.

- g représente la force gravitationnelle.

- ρ_k est la densité du fluide dans la couche k et ρ est la densité moyenne du fluide. Le modèle barocline s'appuie en effet sur le principe de couches de fluide de densité constante.
- h_k est la hauteur de la couche k .
- L'opérateur $\frac{D_k}{Dt}$ correspond à la dérivée particulaire Lagrangienne :

$$\frac{D_k}{Dt} = \frac{\partial}{\partial t} + J(\Psi_k, \cdot)$$

où J est l'opérateur Jacobien

$$J(f, g) = \frac{\partial f}{\partial x} \frac{\partial g}{\partial y} - \frac{\partial f}{\partial y} \frac{\partial g}{\partial x}.$$

- $\alpha \Delta \Psi_n$ représente la dissipation par frottement au fond du bassin et $\beta \Delta^2 \Psi_k$ la dissipation par friction latérale dans chaque couche k .
- Enfin, F_1 est l'unique terme de forçage du modèle océanique. Il correspond à la tension du vent sur la surface du bassin.

5.1.3 Conditions aux bords

La matrice tridiagonale W sert à coupler les fonctions de courant des différentes couches. Elle peut être diagonalisée sous la forme :

$$W = -P \cdot \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_n \end{pmatrix} \cdot P^{-1} \quad (5.4)$$

avec $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$ (la somme des éléments sur une même ligne de W est nulle, donc W est singulière et admet donc 0 pour valeur propre), et P la matrice de passage.

On peut alors définir le vecteur Φ des modes correspondant aux fonctions de courants Ψ :

$$\Phi = \begin{pmatrix} \Phi_1 \\ \vdots \\ \Phi_n \end{pmatrix} := P^{-1} \Psi. \quad (5.5)$$

Les équations (5.1-5.3) peuvent être réécrites suivant les modes :

$$\frac{\partial}{\partial t} (\Delta\Phi - D\Phi) = P^{-1} \begin{pmatrix} -J(\Psi_1, \theta_1(\Psi) + f) + \beta\Delta^2\Psi_1 + F_1 \\ \vdots \\ -J(\Psi_k, \theta_k(\Psi) + f) + \beta\Delta^2\Psi_k \\ \vdots \\ -J(\Psi_n, \theta_n(\Psi) + f) - \alpha\Delta\Psi_n + \beta\Delta^2\Psi_n \end{pmatrix} \quad (5.6)$$

Nous pouvons alors étudier les conditions aux limites. Il faut exprimer le fait que le bassin est fermé. Il y a donc glissement aux bords et conservation de la masse. On a alors :

$$\Psi_k = \text{constante sur } \partial\Omega \times [0, T]. \quad (5.7)$$

Exprimée en termes de modes, cette condition s'écrit :

$$\begin{cases} \Phi_1 = 0 & \text{sur } \partial\Omega \times [0, T], \\ \int_{\Omega} \Phi_k(t) d\sigma = 0 & \forall t \in [0, T], \forall k \geq 2. \end{cases} \quad (5.8)$$

Il faut ajouter l'équation suivante pour compléter les conditions aux bords :

$$\Delta\Psi_k = 0 \quad \text{sur } \partial\Omega \times [0, T], \forall k \in \{1 \dots n\} \quad (5.9)$$

Les équations (5.7-5.9) constituent les conditions aux limites du modèle. Enfin, les conditions initiales (la donnée des $\Psi_k(0)$) s'y ajoutent pour compléter le système des équations du modèle.

5.2 Algorithme 4D-VAR

Nous allons maintenant appliquer l'algorithme d'assimilation variationnelle de données 4D-VAR (voir le chapitre 2 pour les généralités concernant cet algorithme) au modèle quasi-géostrophique barocline dont nous venons de voir les équations. Nous supposons que les données que nous souhaitons assimiler proviennent de mesures satellitaires de la hauteur d'eau à la surface de l'océan. Celle-ci est directement reliée à la fonction de courant de la couche de surface Ψ_1 . Nous allons donc noter Ψ_1^{obs} la fonction de courant observée. Ces observations sont évidemment discrètes en espace et en temps. La fonction Ψ_1^{obs} appartient donc à un espace de dimension plus petite que les fonctions de courant Ψ_k . Notons t_i , $0 \leq i \leq N$, les instants auxquels des observations ($\Psi_1^{obs}(t_i)$) sont disponibles au cours de la période d'assimilation $[0, T]$.

Le vecteur de contrôle, noté u , est la condition initiale du modèle, c'est-à-dire l'ensemble des états initiaux des fonctions de courant de toutes les couches :

$$u = (\Psi_k(0))_{k=1\dots n}. \quad (5.10)$$

5.2.1 Fonction coût

Nous pouvons alors définir une fonction coût mesurant l'écart entre le vecteur de contrôle et les observations du système :

$$\begin{aligned} \mathcal{J}(u) &= \frac{1}{2} \sum_{i=0}^N \langle R_i^{-1} (H_i \Psi_1(t_i) - \Psi_1^{obs}(t_i)), H_i \Psi_1(t_i) - \Psi_1^{obs}(t_i) \rangle \\ &+ \frac{1}{2} \langle P_0^{-1} u, u \rangle. \end{aligned} \quad (5.11)$$

Dans cette équation, les matrices P_0 et R_i sont des matrices de covariance d'erreur. Les opérateurs H_i , opérateurs d'observation reliant les observations $\Psi_1^{obs}(t_i)$ et les solutions du modèle $\Psi_1(t_i)$, sont supposés linéaires. Enfin, $\langle \cdot, \cdot \rangle$ est un produit scalaire.

La première partie de la fonction coût mesure l'écart au sens des moindres carrés entre les observations et la fonction de courant de la couche de surface. La seconde partie est un terme de régularisation. Le problème inverse consistant à minimiser \mathcal{J} est alors bien posé.

5.2.2 Modèle adjoint

Afin de minimiser la fonction coût \mathcal{J} , il est nécessaire de connaître son gradient $\nabla \mathcal{J}$. Comme la dimension du vecteur de contrôle est trop élevée pour calculer la dérivée par une méthode de différences finies, nous allons utiliser la méthode de l'adjoint. Il faut alors résoudre le modèle adjoint de façon rétrograde. Les équations adjointes du modèle quasi-géostrophiques sont :

- pour la couche de surface ($k = 1$) :

$$\begin{aligned} \frac{\partial \theta_1^T(\Lambda)}{\partial t} - \Delta J(\Psi_1, \Lambda_1) - (W^T J(\Psi, \Lambda))_1 \\ - J(\Lambda_1, \theta_1(\Psi) + f) - \beta \Delta^2 \Lambda_1 = E_1 \end{aligned} \quad (5.12)$$

- pour les couches intermédiaires ($2 \leq k \leq n - 1$) :

$$\begin{aligned} \frac{\partial \theta_k^T(\Lambda)}{\partial t} - \Delta J(\Psi_k, \Lambda_k) - (W^T J(\Psi, \Lambda))_k \\ - J(\Lambda_k, \theta_k(\Psi) + f) - \beta \Delta^2 \Lambda_k = 0 \end{aligned} \quad (5.13)$$

– pour la couche du fond ($k = n$) :

$$\begin{aligned} \frac{\partial \theta_n^T(\Lambda)}{\partial t} - \Delta J(\Psi_n, \Lambda_n) - (W^T J(\Psi, \Lambda))_n \\ - J(\Lambda_n, \theta_n(\Psi) + f) + \alpha \Delta \Lambda_n - \beta \Delta^2 \Lambda_n = 0 \end{aligned} \quad (5.14)$$

Les notations utilisées sont les suivantes :

– $\Lambda_1, \dots, \Lambda_n$ sont les fonctions de courant adjointes dans chacune des couches.

– Λ représente le vecteur $\begin{pmatrix} \Lambda_1 \\ \vdots \\ \Lambda_n \end{pmatrix}$.

– $\theta_k^T(\Lambda) = -\Delta \Lambda_k + (W^T \Lambda)_k$ est la vorticit  correspondant   l' tat adjoint.

– Enfin, E_1 est la d riv e de \mathcal{J} par rapport   Ψ_1 :

$$E_1(t) = \sum_{i=0}^N H_i^T R_i^{-1} (H_i \Psi_1(t) - \Psi_1^{obs}(t)) \delta(t - t_i). \quad (5.15)$$

Si on note $\chi = \begin{pmatrix} \chi_1 \\ \vdots \\ \chi_n \end{pmatrix}$ le vecteur modal adjoint :

$$\chi = P^T \Lambda, \quad (5.16)$$

les conditions aux bords que doit satisfaire l' tat adjoint Λ sont :

$$\begin{aligned} \chi_1 = 0 \quad \text{sur } \partial\Omega \times [0, T], \\ \int_{\Omega} \chi_k(t) d\sigma = 0 \quad \forall t \in [0, T], \forall k \geq 2, \end{aligned} \quad (5.17)$$

et

$$\Delta \Lambda_k(t) = 0 \quad \text{sur } \partial\Omega \times [0, T], \forall k. \quad (5.18)$$

5.2.3 Gradient

Le gradient de la premi re partie (not e \mathcal{J}_0) de la fonctionnelle \mathcal{J} s'obtient en r solvant les  quations (5.12-5.14) avec comme condition finale la nullit  des fonctions de courant duales. On obtient alors :

$$\nabla \mathcal{J}_0 = h(-\Delta + W)h^{-1} \begin{pmatrix} \Lambda_1(0) \\ \vdots \\ \Lambda_n(0) \end{pmatrix} \quad (5.19)$$

où h est la matrice diagonale avec les hauteurs h_k des couches sur la diagonale.

Le gradient de la seconde partie de \mathcal{J} s'obtient simplement par dérivation par rapport à u . On a alors :

$$\nabla \mathcal{J} = h(-\Delta + W)h^{-1} \begin{pmatrix} \Lambda_1(0) \\ \vdots \\ \Lambda_n(0) \end{pmatrix} + P_0^{-1}u. \quad (5.20)$$

La minimisation effective de la fonction coût \mathcal{J} se fait en utilisant une méthode L-BFGS.

5.2.4 Erreur modèle

Nous avons jusque là supposé que le modèle était parfait. Ceci n'est pas le cas, puisque certains paramètres sont mal estimés, et les équations simplifiées. Nous avons vu dans le chapitre 2 portant sur les généralités de l'assimilation de données qu'il est impossible de tenir compte de ces erreurs dans le cas de l'algorithme 4D-VAR. En effet, il faudrait rajouter aux équations (5.1-5.3) des termes correctifs pour les modéliser. Ces termes seraient alors à rajouter dans le vecteur de contrôle, et sa taille se verrait multipliée par le nombre de pas de temps utilisés pour l'intégration du modèle. Il ne serait alors plus du tout envisageable de minimiser la fonction coût prenant en compte ces termes correctifs.

On peut malgré tout incorporer de façon partielle l'erreur modèle en la projetant sur une base d'EOF (Empirical Orthogonal Functions) par exemple, et en ne gardant que quelques composantes [53]. La fonction coût ne minimisera alors l'erreur que dans une base assez restrictive.

5.3 Algorithme dual

5.3.1 Généralités

Cet algorithme a été récemment introduit par Bennett [4], Amodei [1] et Courtier [9]. Dans un cadre strictement linéaire, en supposant que le modèle est imparfait, cette nouvelle approche est strictement équivalente au 4D-VAR, mais l'erreur modèle est prise en compte de façon inhérente dans la formulation duale, ce qui évite les problèmes numériques précédemment cités (voir le paragraphe 4D-PSAS dans le chapitre 2).

5.3.2 Description théorique

Méthode primale pour un modèle linéaire

Considérons un modèle linéaire

$$\begin{cases} \frac{dx}{dt} + A(t)x = f + v, \\ x(0) = x_0 + u, \end{cases} \quad (5.21)$$

où A est un opérateur linéaire, v est l'erreur modèle inconnue, et $x_0 + u$ est la condition initiale, également inconnue. x_0 représente une ébauche de la condition initiale, et u est l'erreur correspondante. On suppose comme précédemment que des observations y_i sont disponibles aux instants t_i au cours de la période d'assimilation $[0, T]$.

On peut alors définir une fonction coût type :

$$\begin{aligned} \mathcal{J}(u, v) &= \frac{1}{2} \sum_{i=0}^N \langle R_i^{-1} (H_i x(t_i) - y_i), H_i x(t_i) - y_i \rangle \\ &+ \frac{1}{2} \langle P_0^{-1} u, u \rangle + \frac{1}{2} \int_0^T \langle Q^{-1} v(t), v(t) \rangle dt. \end{aligned} \quad (5.22)$$

où P_0 , Q et R_i désignent encore les matrices de covariance relatives à l'erreur sur la condition initiale, l'erreur modèle et les erreurs d'observation respectivement. H_i représente l'opérateur d'observation linéaire à l'instant t_i permettant de relier l'observation y_i à la solution du modèle au même instant $x(t_i)$.

On peut désormais introduire l'état et le modèle adjoints de façon à calculer le gradient de la fonction coût :

$$\begin{cases} -\frac{dp}{dt} + A(t)^T p = \sum_{i=0}^N H_i^T R_i^{-1} (y_i - H_i x(t_i)) \delta(t - t_i), \\ p(T) = 0, \end{cases} \quad (5.23)$$

et

$$\begin{aligned} \nabla \mathcal{J}(u, v) \cdot \begin{pmatrix} h_1 \\ h_2 \end{pmatrix} &= -\langle p(0), h_1 \rangle - \int_0^T \langle p(t), h_2(t) \rangle dt \\ &+ \langle P_0^{-1} u, h_1 \rangle + \int_0^T \langle Q^{-1} v(t), h_2(t) \rangle dt \end{aligned} \quad (5.24)$$

On a alors le système d'optimalité suivant, d'inconnues (\hat{x}, \hat{p}) :

$$\begin{cases} \frac{d\hat{x}}{dt} + A(t)\hat{x} = f + Q\hat{p}, \\ \hat{x}(0) = x_0 + P_0\hat{p}(0), \end{cases}$$

et

$$\begin{cases} -\frac{d\hat{p}}{dt} + A(t)^T\hat{p} = \sum_{i=0}^N H_i^T R_i^{-1}(y_i - H_i x(t_i)) \delta(t - t_i), \\ \hat{p}(T) = 0. \end{cases}$$

(5.25)

Approche duale

Notons \tilde{x} la solution du modèle direct sans erreur modèle et sans incertitude sur la condition initiale :

$$\begin{cases} \frac{d\tilde{x}}{dt} + A(t)\tilde{x} = f, \\ \tilde{x}(0) = x_0. \end{cases} \quad (5.26)$$

\tilde{x} sera appelée dans la suite solution de référence, et nous considérerons que c'est une ébauche de \hat{x} .

En vue de considérer des équations matricielles, on notera dans la suite y le vecteur comprenant toutes les observations :

$$y = \begin{pmatrix} y_0 \\ \vdots \\ y_N \end{pmatrix}.$$

On notera d_o la dimension d'une observation y_i , le vecteur y étant alors de taille $d_o \times (N + 1)$. On va également définir un opérateur d'observation global H , qui à un état x associe le vecteur

$$Hx = \begin{pmatrix} H_0x(t_0) \\ \vdots \\ H_Nx(t_N) \end{pmatrix}.$$

On note R la matrice diagonale par blocs, avec sur la diagonale les matrices R_i . On définit le vecteur d'innovation

$$d = y - H\tilde{x}$$

comme étant l'écart entre les observations et la solution de référence du problème. On note enfin

$$m = \begin{pmatrix} m_0 \\ \vdots \\ m_N \end{pmatrix}$$

un vecteur quelconque de l'espace des observations. En étudiant le système d'optimalité (5.25), on constate qu'il faut considérer le vecteur d'observation $m = R^{-1}(y - Hx)$.

L'algorithme dual est alors le suivant :

- On choisit un vecteur m dans l'espace des observations, m_i représentant l'observation du système à l'instant t_i .
- On construit alors la solution p_m de l'équation adjointe avec les observations m_i comme forçage aux instants t_i :

$$\begin{cases} -\frac{dp_m}{dt} + A(t)^T p_m = \sum_{i=0}^N H_i^T m_i \delta(t - t_i), \\ p_m(T) = 0. \end{cases} \quad (5.27)$$

- On en déduit alors la condition initiale pour le système direct et on peut résoudre le système suivant :

$$\begin{cases} \frac{dx_m}{dt} + A(t)x_m = Qp_m, \\ x_m(0) = P_0 p_m(0), \end{cases} \quad (5.28)$$

x_m étant alors la solution du problème d'évolution primal avec la condition initiale $P_0 p_m(0)$ et un forçage au second membre égal à $Qp_m(t)$.

- On peut enfin définir le vecteur $\mathcal{D}m = Hx_m$, qui appartient à l'espace des observations.

On a alors la proposition suivante :

Proposition 5.1 *L'opérateur \mathcal{D} agissant sur l'espace des observations est linéaire symétrique défini positif. De plus, si on note \hat{m} la solution de*

$$(\mathcal{D} + R)\hat{m} = d, \quad (5.29)$$

alors

$$\hat{m} = R^{-1}(y - H\hat{x}) \quad \text{et} \quad \hat{x} = \tilde{x} + x_{\hat{m}}. \quad (5.30)$$

■

Démonstration : Notons $M(t, t')$ la résolvante du système associée à l'opérateur $A(t)$: elle permet d'obtenir la solution à l'instant t de l'équation (5.21) en fonction de la solution à l'instant t' . Par exemple, $x(T) = M(T, 0)x(0)$. L'équation (5.28) s'écrit alors

$$x_m(t) = M(t, 0)P_0 p_m(0) + \int_0^t M(t, s) Qp_m(s) ds.$$

Par transposition, on obtient que la résolvante du système associée à $A(t)^T$ est $M(t', t)^T$. Donc

$$p_m(0) = M(T, 0)^T p_m(T) + \int_0^T M(s, 0)^T H_i^T m_i \delta(t - t_i) ds.$$

Choisissons un vecteur m particulier, dont toutes les composantes sont nulles, à l'exception de la $i^{\text{ème}}$ où on trouve la valeur m_i . On a alors, pour ce vecteur m là,

$$p_m(0) = M(t_i, 0)^T H_i^T m_i.$$

On a donc

$$x_m(t) = M(t, 0) P_0 M(t_i, 0)^T H_i^T m_i + \int_0^t M(t, s) Q p_m(s) ds,$$

On a donc

$$\begin{aligned} (\mathcal{D}m)_j &= H_j x_m(t_j) = H_j M(t_j, 0) P_0 M(t_i, 0)^T H_i^T m_i + \int_0^{t_j} H_j M(t_j, s) Q p_m(s) ds \\ &= H_j M(t_j, 0) P_0 M(t_i, 0)^T H_i^T m_i + \int_0^{\min(t_i, t_j)} H_j M(t_j, s) Q M(t_i, s)^T H_i^T m_i ds \end{aligned}$$

car pour $t > t_i$, $p_m(t) = 0$. On en déduit aisément que

$$\mathcal{D}_{ij} = \mathcal{D}_{ji}^T$$

ce qui démontre que \mathcal{D} est symétrique.

Sous l'hypothèse (non restrictive) P_0 et Q symétriques définies positives, l'opérateur \mathcal{D} est alors symétrique non négatif. $\mathcal{D} + R$ est alors un opérateur symétrique défini positif puisque R est symétrique défini positif.

L'existence et l'unicité de \hat{m} en découle directement. La solution optimale \hat{x} du problème primal et la variable adjointe \hat{p} vérifient le système d'optimalité (5.25). On peut donc décomposer \hat{x} sous la forme $\tilde{x} + \underline{x}$, avec \underline{x} la solution de

$$\begin{cases} \frac{d\underline{x}}{dt} + A(t)\underline{x} = Q\hat{p}(t), \\ \underline{x}(0) = P_0\hat{p}(0). \end{cases} \quad (5.31)$$

Puisque $(\mathcal{D} + R)\hat{m} = d$, on a $\mathcal{D}\hat{m} = Hx_{\hat{m}} = d - R\hat{m}$. Or on constate que $x_{\hat{m}}$ et \underline{x} vérifient la même équation d'évolution avec la même condition initiale. Il y a donc égalité, et $\hat{x} = \tilde{x} + x_{\hat{m}}$.

Enfin, par définition,

$$\hat{m} = R^{-1}(d - \mathcal{D}\hat{m}) = R^{-1}((y - H\tilde{x}) - Hx_{\hat{m}}) = R^{-1}(y - H(\tilde{x} + x_{\hat{m}}))$$

ce qui achève la démonstration. \square

Pour résoudre l'équation (5.29), étant donné que $(\mathcal{D} + R)$ est symétrique défini positif, une méthode de gradient permet d'approcher assez rapidement la solution. Il faut donc définir la fonctionnelle quadratique associée, appelée fonctionnelle duale :

$$\mathcal{J}_{\mathcal{D}}(m) = \frac{1}{2} \langle (\mathcal{D} + R)m, m \rangle - \langle d, m \rangle \quad (5.32)$$

où $\langle \cdot, \cdot \rangle$ désigne un produit scalaire sur l'espace des observations.

Lien avec la méthode primale

Nous avons le théorème suivant qui permet de relier les deux approches et de montrer qu'elles sont équivalentes :

Théorème 5.1 *En posant $u_m = x_m(0)$ et $v_m = Qp_m$, on peut redéfinir la fonctionnelle primale \mathcal{J} comme étant une fonction de m uniquement :*

$$\mathcal{J}(m) = \mathcal{J}(u_m, v_m) = \frac{1}{2} \langle \mathcal{D}m, m \rangle + \frac{1}{2} \langle R^{-1}(\mathcal{D}m - d), \mathcal{D}m - d \rangle. \quad (5.33)$$

On a alors

$$\min_m \mathcal{J}(m) = \max_m (-\mathcal{J}_{\mathcal{D}}(m)) = -\min_m \mathcal{J}_{\mathcal{D}}(m) = \frac{1}{2} \langle (\mathcal{D} + R)^{-1}d, d \rangle. \quad (5.34)$$

De plus, les minima sont réalisés au même point m . ■

Démonstration : Commençons par expliciter le second membre de (5.33) :

$$\begin{aligned} \langle \mathcal{D}m, m \rangle &= \sum_{i=0}^N \langle H_i x_m(t_i), m_i \rangle = \int_0^T \langle x_m(t), \sum_{i=0}^N H_i^T m_i \delta(t - t_i) \rangle dt \\ &= \int_0^T \langle x_m(t), -\frac{dp_m}{dt}(t) + A(t)^T p_m(t) \rangle dt \\ &= \int_0^T \langle \frac{dx_m}{dt}(t) + A(t)x_m(t), p_m(t) \rangle dt - \langle x_m(T), p_m(T) \rangle + \langle x_m(0), p_m(0) \rangle \\ &= \int_0^T \langle Qp_m(t), p_m(t) \rangle dt + \langle P_0 p_m(0), p_m(0) \rangle, \end{aligned}$$

et on retrouve les deux derniers termes de \mathcal{J} dans la formule (5.22), compte tenu du fait que $u_m = x_m(0) = P_0 p_m(0)$ et $v_m(t) = Qp_m(t)$.

De plus, $\mathcal{D}m - d = Hx_m - (y - H\tilde{x}) = H(\tilde{x} + x_m) - z$ et $\tilde{x} + x_m$ est solution de l'équation (5.21) avec $u = u_m$ et $v = v_m$. On retombe donc bien sur le premier terme de $\mathcal{J}(u_m, v_m)$.

Dérivons maintenant $\mathcal{J}(m)$:

$$\nabla \mathcal{J}(m) = \mathcal{D}m + \mathcal{D}R^{-1}(\mathcal{D}m - d)$$

donc au minimum m^* de \mathcal{J} , on a

$$\nabla \mathcal{J}(m^*) = 0 \text{ (i.e.) } m^* + R^{-1}(\mathcal{D}m^* - d) = 0 \text{ (i.e.) } R^{-1}(R + \mathcal{D})m^* = R^{-1}d,$$

d'où on tire que

$$\min_m \mathcal{J}(m) = \mathcal{J}(m^*) \text{ avec } m^* = (\mathcal{D} + R)^{-1}d = \hat{m},$$

et

$$\mathcal{J}(\hat{m}) = \frac{1}{2}\langle d - R\hat{m}, \hat{m} \rangle + \frac{1}{2}\langle \hat{m}, R\hat{m} \rangle = \frac{1}{2}\langle d, \hat{m} \rangle = \frac{1}{2}\langle (\mathcal{D} + R)^{-1}d, d \rangle.$$

Par construction de $\mathcal{J}_{\mathcal{D}}$, on a clairement

$$\min_m \mathcal{J}_{\mathcal{D}}(m) = \mathcal{J}_{\mathcal{D}}(\hat{m}) = -\frac{1}{2}\langle d, \hat{m} \rangle = -\frac{1}{2}\langle (\mathcal{D} + R)^{-1}d, d \rangle. \quad \square$$

Le théorème 5.1 démontre l'équivalence des deux approches en terme de construction des solutions, ainsi que l'égalité des solutions construites et des extréma des fonctionnelles. Il y a cependant deux avantages nets en faveur de la méthode duale :

- la prise en compte inhérente de l'erreur modèle,
- et la minimisation de la fonction coût duale se fait dans l'espace des observations, de dimension inférieure à l'espace des états pour la fonctionnelle primale.

5.3.3 Algorithme dual pour le modèle quasi-géostrophique

Voyons maintenant comment l'algorithme dual peut être utilisé dans le cas d'un modèle non linéaire, le modèle quasi-géostrophique barocline. Nous rappelons qu'un des avantages de la méthode duale est qu'elle couvre aussi la méthode primale dans un cadre linéaire. En effet, la méthode duale agit sur les matrices de covariance d'erreur directes et en faisant tendre la matrice de covariance d'erreur modèle vers zéro, la méthode n'est pas perturbée et devient équivalente à la méthode primale pour un problème sans erreur modèle.

Du fait de la non linéarité du modèle, la trajectoire utilisée pour linéariser le modèle autour d'un état de référence évolue ici à chaque itération, et nous avons donc abandonné l'écriture sous forme incrémentale $\hat{x} = \tilde{x} + x_{\hat{m}}$. Nous devons donc construire un *nouvel* algorithme dual pour notre modèle qui consiste à calculer une fonction coût

duale de la fonctionnelle primale (comme indiqué dans le paragraphe 2.2 du chapitre 2) via une linéarisation adéquate autour d'un état de référence. La difficulté majeure de l'algorithme dual vient du fait que la trajectoire utilisée pour linéariser le modèle est réactualisée à chaque itération, et par conséquent, les opérateurs d'observation linéarisés et \mathcal{D} aussi. La fonction coût duale change donc aussi à chaque itération dans le processus de minimisation.

L'algorithme dans le cas non linéaire se présente sous la forme suivante :

- Soit m un vecteur d'observations qui peut être relié directement à Ψ_1 (supposons par exemple que m contient les observations d'une partie de la surface de l'océan à différents instants t_i). Donnons nous un état de référence du modèle (une ébauche) afin de pouvoir linéariser le modèle et les opérateurs d'observation.
- On résout les équations adjointes, provenant d'une linéarisation du modèle autour de la trajectoire issue de l'ébauche :

$$\begin{aligned} \frac{\partial \theta_1^T(\Lambda)}{\partial t} - \Delta J(\Psi_1, \Lambda_1) - (W^T J(\Psi, \Lambda))_1 - J(\Lambda_1, \theta_1(\Psi) + f) \\ - \beta \Delta^2 \Lambda_1 = \tilde{E}_1(m), \\ \frac{\partial \theta_k^T(\Lambda)}{\partial t} - \Delta J(\Psi_k, \Lambda_k) - (W^T J(\Psi, \Lambda))_k - J(\Lambda_k, \theta_k(\Psi) + f) \\ - \beta \Delta^2 \Lambda_k = 0, \quad 1 < k < n, \end{aligned} \quad (5.35)$$

$$\begin{aligned} \frac{\partial \theta_n^T(\Lambda)}{\partial t} - \Delta J(\Psi_n, \Lambda_n) - (W^T J(\Psi, \Lambda))_n - J(\Lambda_n, \theta_n(\Psi) + f) \\ + \alpha \Delta \Lambda_n - \beta \Delta^2 \Lambda_n = 0, \end{aligned}$$

avec les conditions finales $\Lambda_k(T) = 0$, $1 \leq k \leq n$, et

$$\tilde{E}_1(m)(t) = \sum_{i=0}^N H_i^T R_i^{-1} (m(t) - \Psi_1^{obs}(t)) \delta(t - t_i).$$

- Il faut ensuite résoudre les équations directes non linéaires :

$$\begin{aligned} \frac{D_1(\theta_1(\Psi) + f)}{Dt} - \beta \Delta^2 \Psi_1 = F_1 + (Q\Lambda)_1, \\ \frac{D_k(\theta_k(\Psi) + f)}{Dt} - \beta \Delta^2 \Psi_k = (Q\Lambda)_k, \quad 1 < k < n, \\ \frac{D_n(\theta_n(\Psi) + f)}{Dt} + \alpha \Delta \Psi_n - \beta \Delta^2 \Psi_n = (Q\Lambda)_n, \end{aligned} \quad (5.36)$$

avec les conditions initiales

$$\Psi_k(0) = \Psi_k^e(0) + (P_0\Lambda(0))_k, \quad (5.37)$$

où Q et P_0 sont des matrices de préconditionnement statistique et $\Psi_k^e(0)$ est une estimation a priori de l'état initial $\Psi_k(0)$.

La trajectoire obtenue servira de nouvelle ébauche du système pour la prochaine itération.

- On définit alors l'opérateur \mathcal{D} (dépendant de l'état autour duquel a été linéarisé le problème) agissant sur l'espace des observations :

$$(\mathcal{D}m)(t) = \sum_{i=0}^N H_i \Psi_1(t_i) \delta(t - t_i), \quad (5.38)$$

les opérateurs d'observation H_i ayant été linéarisés avec le modèle.

Une fois l'opérateur \mathcal{D} construit, il ne reste plus qu'à utiliser la théorie vue précédemment pour définir la fonction coût duale :

$$\mathcal{J}_{\mathcal{D}}(m) = \frac{1}{2} \langle \mathcal{D}m, m \rangle - \langle \Psi_1^{obs}, m \rangle. \quad (5.39)$$

La fonctionnelle $\mathcal{J}_{\mathcal{D}}$ mesure l'écart entre $\mathcal{D}m$ et Ψ_1^{obs} , c'est-à-dire entre la trace dans l'espace des observations de la solution du modèle direct et le vecteur des observations du système.

Comme l'opérateur \mathcal{D} est linéaire symétrique défini positif, la gradient de $\mathcal{J}_{\mathcal{D}}$ est immédiat :

$$\nabla \mathcal{J}_{\mathcal{D}}(m) = \mathcal{D}m - \Psi_1^{obs}. \quad (5.40)$$

Nous avons maintenant tout ce qu'il faut pour appliquer un algorithme de minimisation de la fonction coût duale $\mathcal{J}_{\mathcal{D}}$, en utilisant simplement une méthode itérative de type quasi-Newton comme par exemple un algorithme L-BFGS, et en réactualisant à chaque itération la fonction coût duale avec la nouvelle ébauche obtenue.

En supposant le minimum \hat{m} identifié, il est alors facile de reconstruire la trajectoire correspondante dans l'espace des états : il suffit de résoudre les équations (5.35) et (5.36), la condition initiale dans l'espace des états étant donné par la formule (5.37).

On observe à nouveau que la fonctionnelle duale est définie sur un espace (celui des observations) de dimension plus petite que la fonctionnelle primale (définie sur l'espace des états). La minimisation de la fonctionnelle duale ne coûte donc pas plus cher que la minimisation de la fonctionnelle primale. De plus, cet algorithme tient toujours compte de l'erreur modèle, ce qui était numériquement impossible avec l'approche primale du 4D-VAR.

Chapitre 6

Résultats numériques

Les méthodes primale (4D-VAR) et duale (algorithme dérivé du 4D-PSAS que nous venons juste de voir) ont été appliquées numériquement au modèle quasi-géostrophique barocline. Nous allons maintenant voir les différents résultats et comparaisons numériques.

6.1 Paramètres numériques du modèle

6.1.1 Valeurs des coefficients du modèle

Les expériences numériques ont été réalisées sur un océan carré à trois couches. Les dimensions horizontales du bassin sont de 4000 km par 4000 km. Sa profondeur est de 5 km. Les profondeurs des différentes couches sont respectivement 300 mètres pour la couche de surface, 700 mètres pour la couche intermédiaire et 4000 mètres pour la couche du fond. L'océan est discrétisé suivant un pas horizontal d'espace de 20 km. La grille de discrétisation a donc $201 \times 201 \times 3$ nœuds. Le pas de temps pour les intégrations numériques des équations différentielles est de 5400 secondes, soit une heure et demie. Les conditions initiales $\Psi_k(0)$ sont choisies égales à zéro pour une phase de spin-up de 6 ans. Après cela, la période d'assimilation commence (correspondant au temps $t = 0$) et dure 5 jours ($t = T$), soit 80 pas de temps.

Les valeurs des autres paramètres numériques du modèle sont récapitulées dans le tableau 6.1.

6.1.2 Paramètres des minimisations

La méthode numérique utilisée pour la minimisation des fonctions coût est un algorithme de type quasi-Newton, BFGS à mémoire limitée. Le code M1QN3 développé par Gilbert et Lemaréchal [19] a été utilisé dans nos expériences.

Gravité réduite dans chacune des couches	10 - 0.0357 - 0.0162 $m.s^{-2}$
Paramètre de Coriolis de référence	9.3e-5 s^{-1}
Gradient méridional du paramètre de Coriolis	2.e-11 $m^{-1}.s^{-1}$
Coefficient de diffusion latérale	1.e+9 s^{-1}
Coefficient de dissipation par frottement au fond	1.e-7 $m^2.s^{-1}$
Tension maximale du vent en surface	1.e-4 s^{-2}

TAB. 6.1 – Paramètres du modèle quasi-géostrophique barocline.

Le nombre maximal d'itérations a été fixé à 40 puisque la taille du problème ne permet pas d'obtenir le minimum absolu des fonctionnelles et qu'en vue d'une méthode opérationnelle d'assimilation de données, le temps de calcul nous est compté. Le nombre de simulations (évaluations de la fonction coût et de son gradient) est alors fixé à 48.

Les autres paramètres ont été fixés comme à la fin du chapitre 3 : la formule utilisée pour mettre à jour le préconditionneur diagonal de l'algorithme BFGS est la formule BFGS directe. La mise à l'échelle de ce préconditionneur n'a été effectuée que pour la minimisation et pas pour sa mise à jour. Le nombre de paires stockées pour la version à mémoire limitée de l'algorithme BFGS est de 5, et ce sont les paires les plus récentes qui sont conservées.

6.2 Solution exacte, observations bruitées

L'approche expérimentale consiste à réaliser des expériences jumelles avec des données simulées. Tout d'abord, une expérience servant de référence est réalisée et les données correspondant à cette trajectoire sont récupérées. On assimilera cette trajectoire à la solution exacte de notre problème. Les données ainsi récupérées ne concernent que la couche de surface de l'océan, généralement (sauf précision contraire) tous les 5 pas d'espace sur la grille et tous les 5 pas de temps, c'est-à-dire toutes les 7 heures et demie. Les données de surface ainsi simulées sont ensuite bruitées à l'aide d'une distribution gaussienne de moyenne nulle, et sont utilisées après comme observations pour l'assimilation. Les résultats du processus d'identification peuvent alors être comparés à l'expérience de référence pour tester la validité des méthodes utilisées. La matrice P_0 a été choisie égale à l'identité, afin de décorréler les erreurs sur la condition initiale.

6.2.1 Spin-up

La phase de spin-up est nécessaire pour que l'océan rentre dans une phase de régime permanent. En effet, en prenant comme condition initiale des fonctions de courant nulles, l'océan est au repos, et il faut une certaine période avant que les courants de bord ouest et la recirculation océanique se stabilisent dans une sorte de régime permanent. Dès quelques mois, un jet apparaît sur le bord ouest, et au bout de 5 années, celui-ci semble avoir atteint sa taille maximale. On peut donc considérer qu'à partir de 5 ans, la phase de spin-up est terminée. Nous avons donc attendu 6 ans avant de commencer les expériences jumelles.

6.2.2 Trajectoire de référence

L'état de l'océan au bout de la phase de spin-up de 6 ans, instant qui nous sert dans la suite d'instant initial de la période d'assimilation, c'est-à-dire $t = 0$, a été pris comme condition initiale exacte. Le modèle quasi-géostrophique barocline a ensuite été intégré pendant 5 jours, soit 80 pas de temps. La longueur de la période d'assimilation a été choisie égale à 5 jours ($T = 5$ jours).

La figure 6.1 représente les trois fonctions de courant Ψ_1 , Ψ_2 et Ψ_3 (relatives aux 3 couches du modèle) au début et à la fin de la période d'assimilation. Ces champs serviront dans la suite à mesurer la qualité de l'identification de l'état initial (fig. 6.1-a) ainsi que la reconstruction des fonctions de courant à l'instant final $t = T$ en vue de prévisions.

Dans la suite, l'ébauche servant à initialiser les algorithmes de minimisation est l'état de référence de l'océan 10 jours avant le début de la période d'assimilation. La couche de surface de l'ébauche est représentée sur la figure 6.2.

6.2.3 Bruitage des observations

Des observations ont été récupérées tous les 5 pas de temps durant toute la période d'assimilation de données. Il y en a donc à 17 instants au total. Elles ne concernent qu'une partie de la couche de surface de l'océan, tous les 5 pas d'espace sur la grille horizontale. Chacun des 17 vecteurs d'observation comprend donc 41×41 éléments, contre 201×201 pour un vecteur représentant une fonction de courant d'une couche, celle de surface par exemple.

Les observations sont alors bruitées avec un bruit blanc (suivant une distribution gaussienne) à hauteur de 10% environ. Ce sont ces observations bruitées qui serviront à l'assimilation de données. La matrice de covariance d'erreur d'observation a été fixée de façon cohérente avec le bruit introduit.

La figure 6.3-a montre les trois niveaux de l'océan une fois bruité (les observations bruitées sont des extraits de la première couche). Les courbes de niveau sont

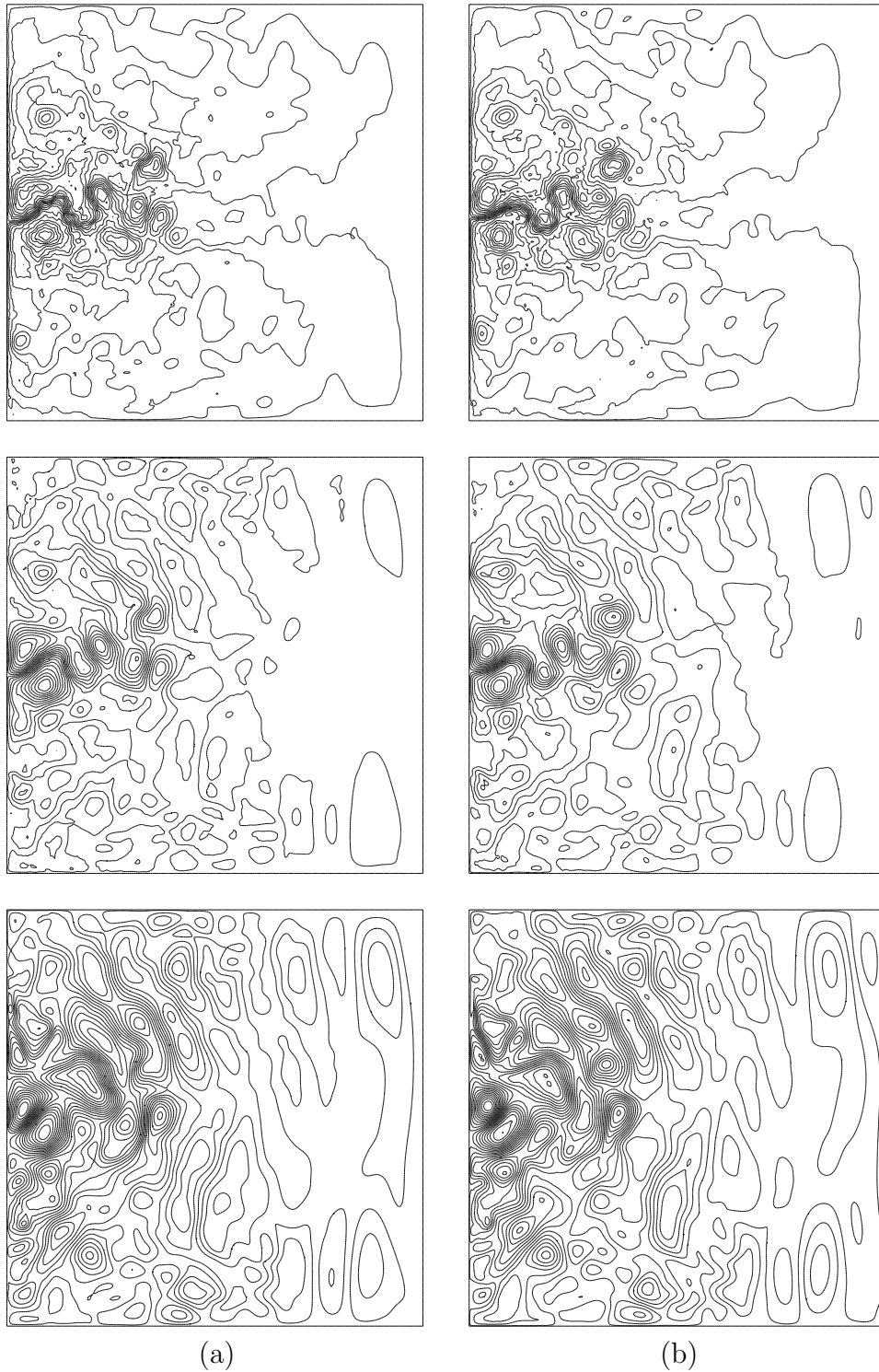


FIG. 6.1 – État initial exact (a) et solution exacte à l'instant final de la période d'assimilation (b) des trois couches de l'océan.

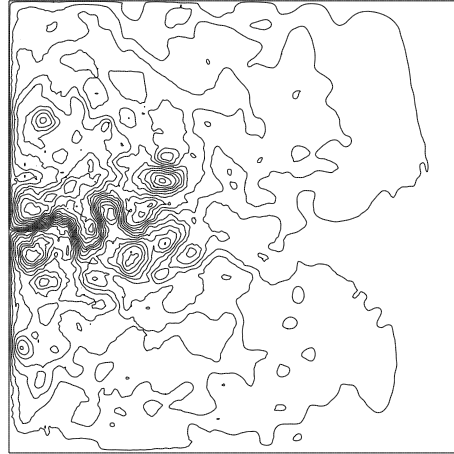


FIG. 6.2 – Couche de surface de l'ébauche utilisée pour les algorithmes de minimisation.

alors tellement bruitées qu'il est parfois impossible de visualiser certains tourbillons, notamment sur la couche de surface, dont les contours étaient parfaitement visibles avant bruitage sur la figure 6.1-a.

La figure 6.3-b montre l'état du système après intégration du modèle sur la période d'assimilation avec l'océan bruité (fig. 6.3-a) pour condition initiale. Une fois encore, de nombreux tourbillons (visibles sur la figure 6.1-b, sans aucun bruitage) ne sont plus identifiables. Seul le jet reste à peu près identifiable.

Ceci montre tout l'intérêt de l'assimilation de données car, en supposant qu'il y ait suffisamment d'observations disponibles pour couvrir toute la grille de discrétisation de l'océan, le bruitage est tel qu'aucune prévision réaliste n'est possible, même à court terme. Le modèle ne va en effet pas lisser la trajectoire, et ce n'est donc pas possible de faire des prévisions simplement en intégrant le modèle à partir d'observations.

6.3 Convergence de la méthode duale et comparaison avec la méthode primale

6.3.1 Convergence de la méthode duale

Nous allons commencer par vérifier que la méthode duale converge numériquement. Nous avons utilisé l'ébauche pour initialiser la minimisation de la fonctionnelle duale, et nous avons utilisé les observations non bruitées.

La figure 6.4 montre l'évolution de la fonction coût duale (a) et de son gradient (b) en fonction du nombre d'itérations dans l'algorithme de minimisation. En une vingtaine d'itérations, nous avons gagné environ 2 ordres de grandeur sur la fonction

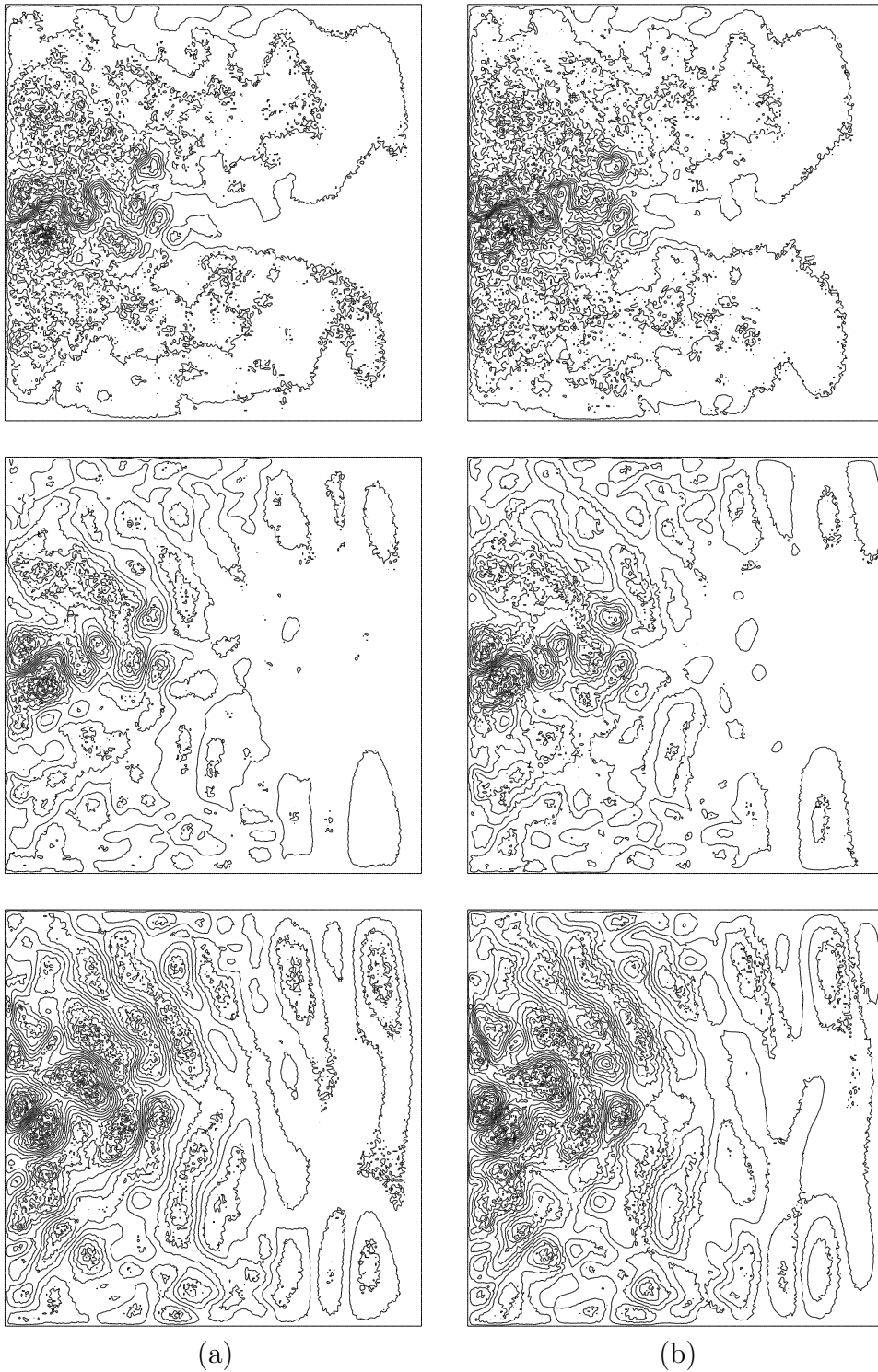


FIG. 6.3 – Données bruitées des différentes couches de l’océan dont seront extraites les observations (a) et état correspondant à la fin de la période d’assimilation après intégration du modèle (b).

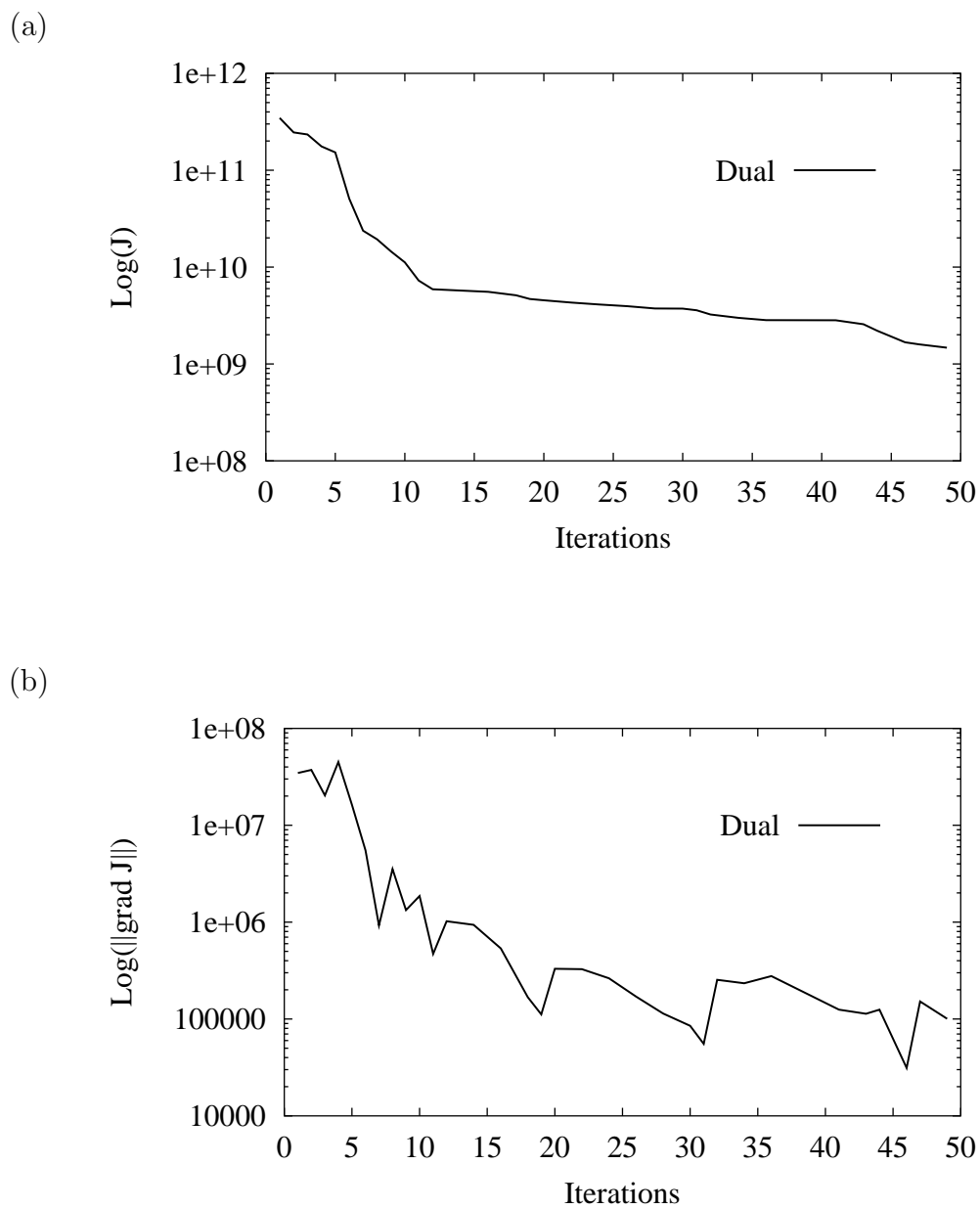


FIG. 6.4 – Évolution de la fonction coût duale (a) et de son gradient (b) en fonction du nombre d'itérations dans l'algorithme de minimisation.

coût, puis celle-ci diminue moins vite avec les itérations. Le gradient est quant à lui diminué d'environ 3 ordres de grandeur avant de se stabiliser. La convergence mathématique n'est pas acquise, mais numériquement, la diminution de la fonction coût duale et de son gradient sont suffisantes pour considérer que la méthode converge.

6.3.2 Comparaison avec la méthode primale sur un problème de faible dimension

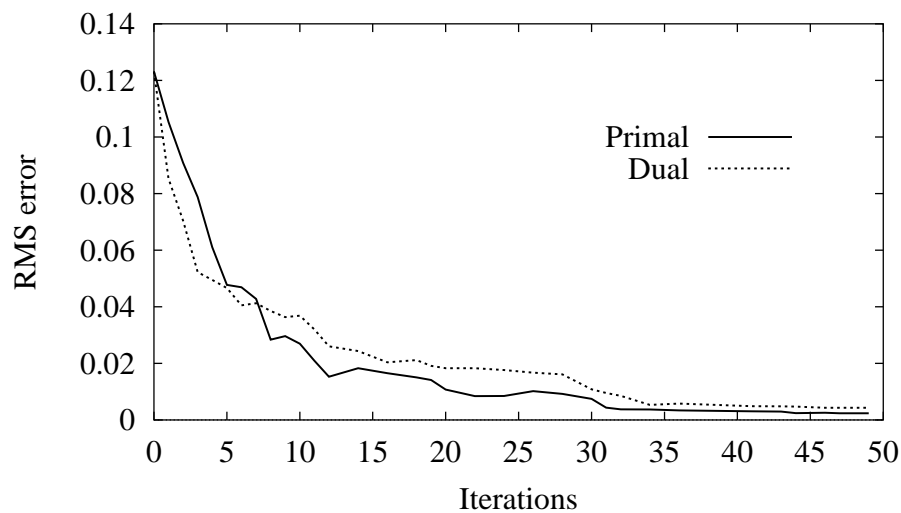
Dans cette partie (uniquement), nous avons réduit la dimension de notre problème afin d'étudier sur un problème de petite taille la perte d'équivalence entre les deux méthodes en passant du linéaire au non linéaire. Nous avons réduit la grille de discrétisation spatiale pour la ramener à une dimension de $41 \times 41 \times 3$, soit un pas horizontal d'espace de 100 km. La dimension de l'espace de contrôle est alors de 5043. Après une phase de spin-up de 15 ans, la période d'assimilation débute et dure toujours 80 pas de temps, soit 5 jours. L'ébauche considérée ici est l'état réel du système 10 jours avant le début de la période d'assimilation. Nous avons récupéré des observations à partir de la trajectoire exacte tous les 5 pas de temps et d'espace. Nous ne les avons pas bruitées. La dimension de l'espace d'observation est alors de 1377.

Dans un premier temps, nous avons comparé les deux méthodes sur le modèle linéarisé. Les fonctionnelles primale et duale sont alors strictement quadratiques, les modèles utilisés lors de la minimisation sont le modèle linéaire tangent et le modèle adjoint du linéaire tangent. Les méthodes primale et duale sont donc mathématiquement équivalentes et doivent conduire en théorie au même minimum. Numériquement, et en arrêtant les minimisations après 50 itérations pour les deux méthodes, les résultats diffèrent évidemment puisqu'aucun des deux algorithmes n'a eu le temps de converger vers le minimum, seulement de s'en approcher.

La figure 6.5-a montre la différence en norme RMS entre l'état initial reconstitué et l'état initial exact, pour chaque méthode, en fonction du nombre d'itérations dans la minimisation. On constate que même si la méthode duale fournit dès les premières itérations un meilleur résultat, assez vite, la méthode primale devient meilleure, pour le rester jusqu'à la 50^{ème} itération. Celle-ci fournit globalement de meilleurs résultats que la méthode duale dans un cadre purement linéaire.

Nous avons ensuite appliqué les deux algorithmes sur le modèle non linéaire, en partant de la même ébauche que précédemment. Cette fois-ci, les méthodes ne sont plus mathématiquement équivalentes. La fig 6.5-b montre dans ce cas la différence en norme RMS entre l'état initial identifié par chacune des deux méthodes d'assimilation et l'état initial exact en fonction du nombre d'itérations dans l'algorithme de minimisation. Nous constatons que la méthode primale divise très rapidement par deux l'erreur sur la condition initiale avant de se stabiliser temporairement. La

(a)



(b)

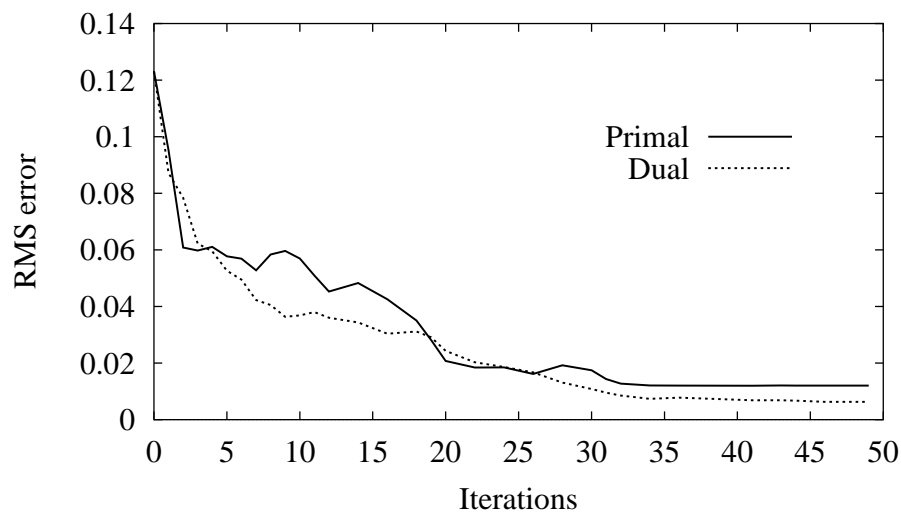


FIG. 6.5 – Évolution de l'erreur sur l'identification de la condition initiale en fonction du nombre d'itérations dans la minimisation, pour le problème linéarisé (a) et pour le problème non linéaire (b).

méthode duale fournit globalement de meilleurs résultats que la méthode primale dans le cadre non linéaire. La différence des résultats entre le modèle linéarisé et le modèle non linéaire s'explique par la perte d'équivalence des méthodes dans un cadre non linéaire et la méthode duale semble alors fournir de meilleurs résultats que la méthode primale.

6.4 Reconstitution de l'état initial et prévisions

Nous allons maintenant essayer de reconstituer l'état initial de référence (fig. 6.1-a) à partir des observations bruitées, en utilisant soit la méthode primale, reposant sur l'algorithme 4D-VAR, soit la méthode duale.

À partir de l'état initial reconstitué (par l'une ou l'autre des méthodes d'assimilation), on étudiera aussi la qualité de l'état final correspondant à la fin de la période d'assimilation et qui sert de début de phase de prévisions.

6.4.1 Méthode primale

Nous rappelons que le vecteur de contrôle u est l'ensemble des conditions initiales des fonctions de courant des 3 couches. Sa dimension est donc $201 \times 201 \times 3$ soit plus de 10^5 . La fonction coût que nous voulons minimiser est donnée par l'équation (5.11). Son gradient est obtenu par la méthode de l'adjoint au premier ordre, et la fonctionnelle est minimisée grâce à un algorithme L-BFGS. La minimisation est arrêtée après 40 itérations, et au plus 48 simulations (le programme de minimisation autorise 20% de simulations supplémentaires par rapport au nombre maximal d'itérations), chaque simulation comprenant une intégration du modèle direct (pour calculer \mathcal{J}) et une intégration rétrograde du modèle adjoint (pour calculer $\nabla \mathcal{J}$).

Le résultat de la minimisation est montré sur la figure 6.6-a. Le modèle direct est ensuite intégré le long de la période d'assimilation en utilisant l'état initial reconstitué comme condition initiale. L'état correspondant du système à l'instant final est représenté sur la figure 6.6-b.

On constate que la fonction de courant de la couche de surface obtenue à l'instant initial est comparable à celle de la solution exacte au même instant (fig. 6.1-a). Ceci est déjà moins vrai à l'instant final $t = T$ (figures 6.6-b et 6.1-b).

Enfin, les couches intermédiaire et au fond sont tout de même bien reconstruites, malgré l'absence d'informations disponibles. Ce phénomène a quasiment toujours été observé et l'information récupérée à partir des observations sur la couche de surface a tendance à se propager aux couches inférieures grâce au couplage entre les différentes couches dans le modèle ([35], [34]).

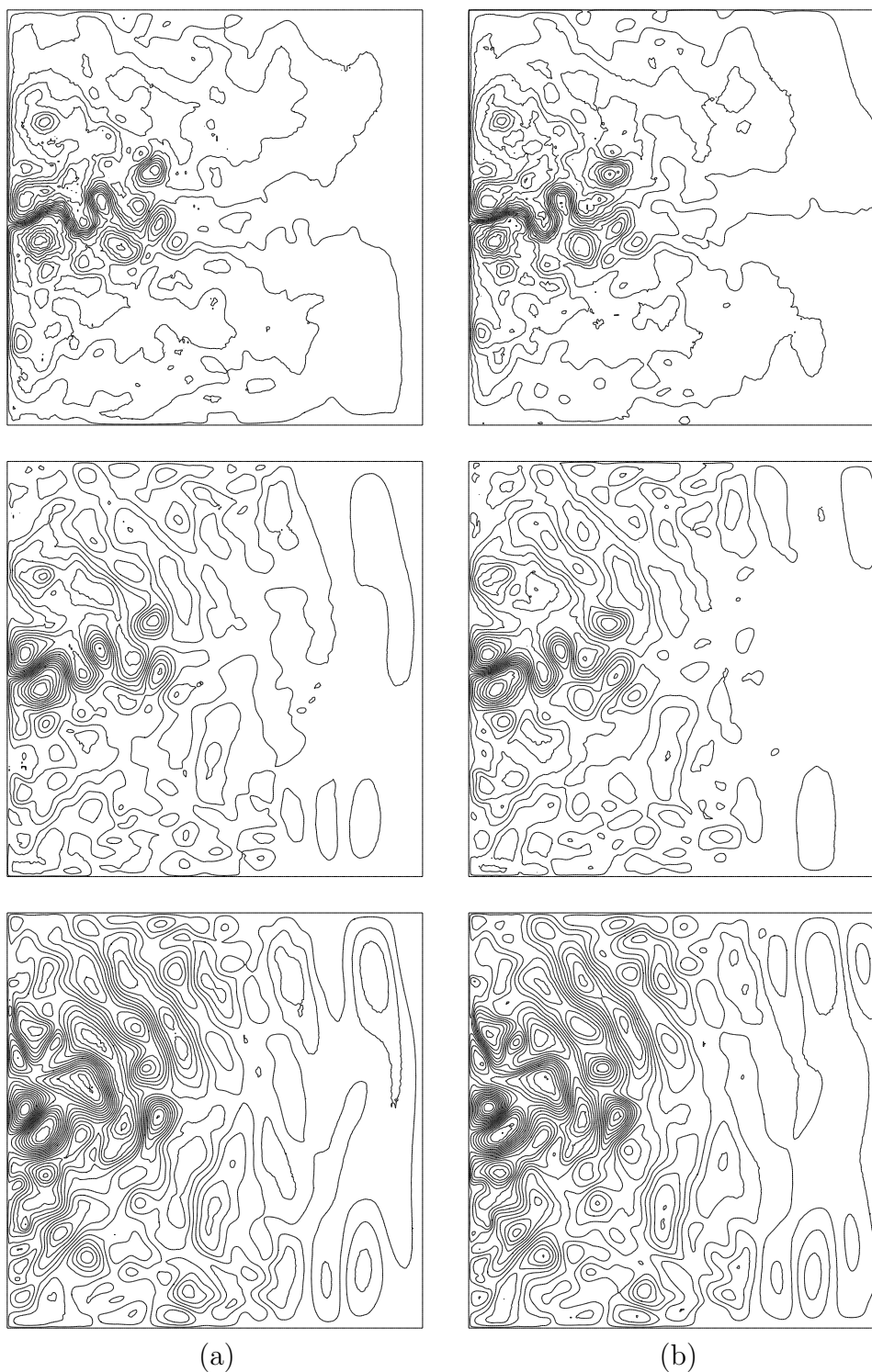


FIG. 6.6 – Résultat de la minimisation de la fonction coût primale. Solution au début (a) et à la fin (b) de la période d'assimilation.

6.4.2 Méthode duale

L'estimation de la condition initiale est la même que pour l'algorithme primal. La minimisation est toujours arrêtée après 40 itérations et au plus 48 simulations, chaque simulation comprenant désormais une intégration rétrograde du modèle adjoint puis une intégration du modèle direct afin de calculer $\mathcal{J}_{\mathcal{D}}$ et $\nabla \mathcal{J}_{\mathcal{D}}$. Le point de départ de la minimisation de la fonctionnelle duale est le vecteur regroupant toutes les observations (bruitées) du système récupérées tout au long de la période d'assimilation. La dimension du vecteur de contrôle dual est de $41 \times 41 \times 17$ (il y a 17 instants où des observations sont disponibles), c'est-à-dire 4 à 5 fois plus petit que le vecteur de contrôle primal.

Le résultat de la minimisation est représenté sur la figure 6.7-a à l'instant initial $t = 0$ et sur la figure 6.7-b à l'instant final $t = T$.

Les fonctions de courant semblent moins lisses que dans le cas de la méthode primale. En effet, les observations sont fortement bruitées, et l'algorithme dual travaille dans l'espace des observations, le point de départ de la minimisation étant les observations bruitées représentées sur la figure 6.3-a. Il semble alors naturel qu'un vecteur d'observations qui suit à peu près les données tout au long de la période d'assimilation reste tout de même bruité.

L'état du système obtenu à la fin de la période d'assimilation est plus proche de la solution exacte à $t = T$ que dans le cas de la méthode primale. Enfin, ici aussi, les fonctions de courant des couches du fond sont bien identifiées malgré le fait qu'aucune information ne les concernant n'était disponible.

6.4.3 Comparaison

La figure 6.8 représente l'erreur mesurée en norme RMS (root mean square), c'est-à-dire au sens des moindres carrés, tout au long de la période d'assimilation, entre la trajectoire de référence (considérée comme solution exacte) et la solution identifiée à l'aide d'une des méthodes précédemment vues (en utilisant directement la première observation bruitée, avec l'algorithme primal ou avec l'algorithme dual). L'erreur RMS à l'instant t pour chacune de ces solutions est mesurée de la façon suivante :

$$rms(t) = \frac{\int_{\Omega} [\Psi_1^{sol}(t) - \Psi_1^{exacte}(t)]^2 d\sigma}{\int_{\Omega} [\Psi_1^{exacte}(t)]^2 d\sigma}. \quad (6.1)$$

L'erreur atteint environ 10% dans le cas de la trajectoire provenant directement des observations, et augmente avec le temps. Ceci est en accord avec les non linéarités inhérentes du modèle qui ont tendance à propager les erreurs en les augmentant au fil du temps.

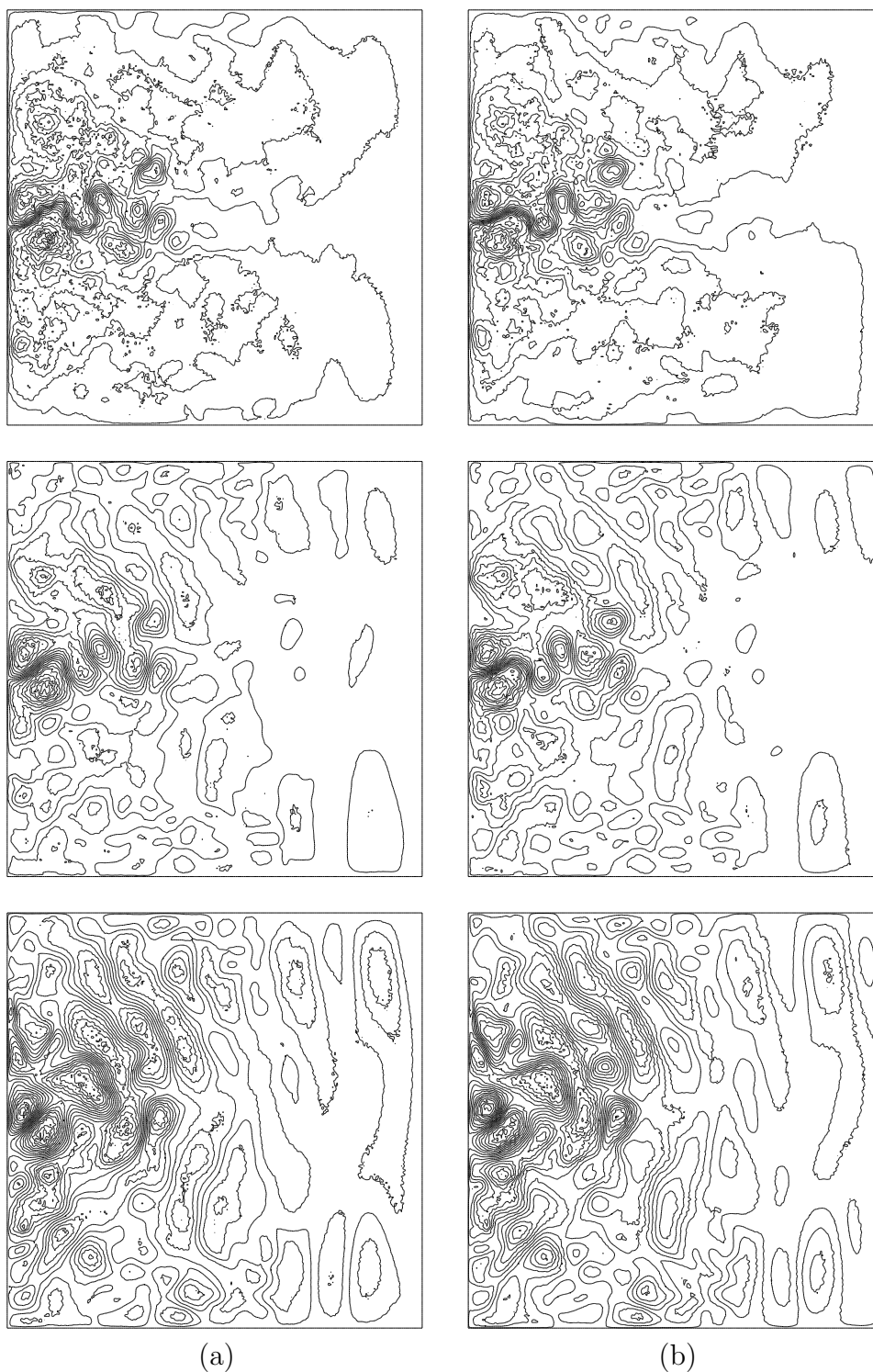


FIG. 6.7 – Résultat de la minimisation de la fonction coût duale. Solution au début (a) et à la fin (b) de la période d'assimilation.

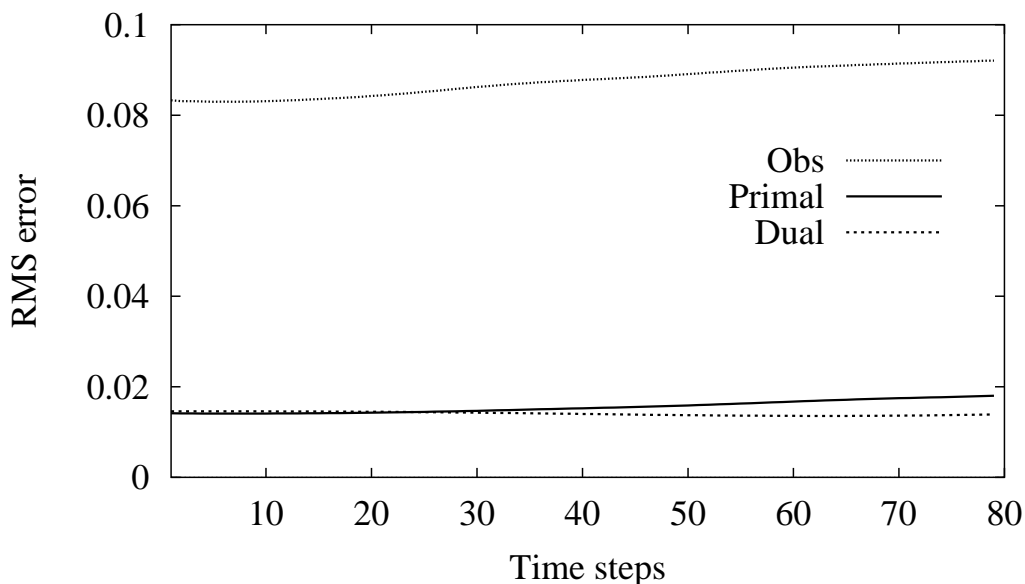


FIG. 6.8 – Normes RMS des erreurs obtenues pour les différentes méthodes en fonction du nombre de pas de temps le long de la période d’assimilation. Utilisation de la première observation bruitée (pointillés rapprochés), algorithme primal (ligne continue) et algorithme dual (ligne pointillée).

Dans le cas de l’utilisation de l’une ou l’autre des deux méthodes d’assimilation de données, l’erreur est clairement plus petite, au moins 5 fois plus faible. Ceci prouve une fois encore l’intérêt des méthodes d’assimilation de données, qui permettent de reconstruire avec moins de 2% d’erreur une trajectoire en utilisant uniquement des observations partielles bruitées à hauteur de 10%.

L’erreur de la solution primale a tendance à augmenter avec le temps. Ceci est cohérent avec la remarque formulée précédemment au sujet de la ressemblance entre la trajectoire reconstruite et la trajectoire de référence surtout au début de la période d’assimilation et moins à la fin.

L’erreur d’assimilation commise avec la méthode duale est un peu plus grande au début de la période d’assimilation, mais elle reste globalement constante (en fait elle diminue très légèrement vers le 60^{ème} pas de temps avant de réaugmenter tout aussi faiblement vers les derniers pas de temps) avec le nombre de pas de temps. Ceci est également cohérent avec la remarque formulée à la fin du paragraphe concernant la méthode duale, qui a mieux reconstitué la trajectoire à l’instant final que la méthode primale.

Il faut également noter que, bien que les données ne concernent que la couche en surface, la couche intermédiaire et la couche du fond du bassin sont aussi bien reconstituées dans un cas comme dans l’autre. Ceci tend à confirmer le fait que les

méthodes d'assimilation de données se servent du modèle pour propager l'information, et par le couplage que le modèle effectue entre les différentes couches, l'information est effectivement propagée de la surface aux autres couches.

Enfin, la dimension du vecteur de contrôle dual (28577) étant plus faible que celle du vecteur de contrôle primal (121203), même si la minimisation est arrêtée pour le même nombre maximal d'itérations/simulations, la minimisation de la fonctionnelle duale est un peu plus poussée, la norme du gradient ayant plus diminué que pour la fonctionnelle primale. En ne s'autorisant qu'environ une trentaine d'itérations lors de la minimisation de la fonctionnelle duale, on obtient une décroissance proche de celle obtenue pour la fonctionnelle primale avec 40 itérations autorisées. Il est donc possible avec la méthode duale de gagner un peu de temps de calcul en conservant des résultats comparables, ou alors garder le même temps de calcul et obtenir des résultats un peu meilleurs qu'avec l'algorithme primal.

6.5 Sensibilité à un terme d'erreur modèle au cours de l'assimilation

Nous allons maintenant bruiteur le modèle en rajoutant un terme supplémentaire dans les équations du modèle. En effet, un des intérêts de la méthode duale par rapport à la méthode primale est la prise en compte de l'erreur modèle, chose qui est numériquement impossible dans l'approche primale. Nous allons donc rajouter un terme d'erreur modèle dans chacune des équations du modèle direct (5.1-5.3). Il s'agit d'un bruit blanc gaussien, biaisant le modèle à hauteur de 2 ou 5%. Il faut noter qu'il a été impossible de mettre plus de 7% d'erreur modèle sans faire diverger de temps en temps le modèle. Les observations que nous assimilons sont les mêmes que dans les paragraphes précédents.

6.5.1 Méthode primale

Nous avons tout d'abord appliqué la méthode primale afin d'assimiler les observations (fig. 6.3) bruitées. La figure 6.9 montre la solution obtenue au début et à la fin de la période d'assimilation dans le cas d'une erreur modèle de 2%. Nous constatons que les lignes de niveau restent lisses malgré le terme d'erreur dans les équations du modèle. En comparaison de la figure 6.6 qui montrait la solution de la méthode primale sans erreur modèle, on constate une perte de qualité de l'identification de l'état initial. Celle-ci se retrouve naturellement sur la reconstitution de l'état final.

La figure 6.10 montre les mêmes résultats que la figure précédente lorsque l'erreur modèle est augmentée et atteint 5%. Il apparaît assez clairement que la reconstitution de l'état initial (et l'état final correspondant) est très nettement dégradée. Néanmoins,

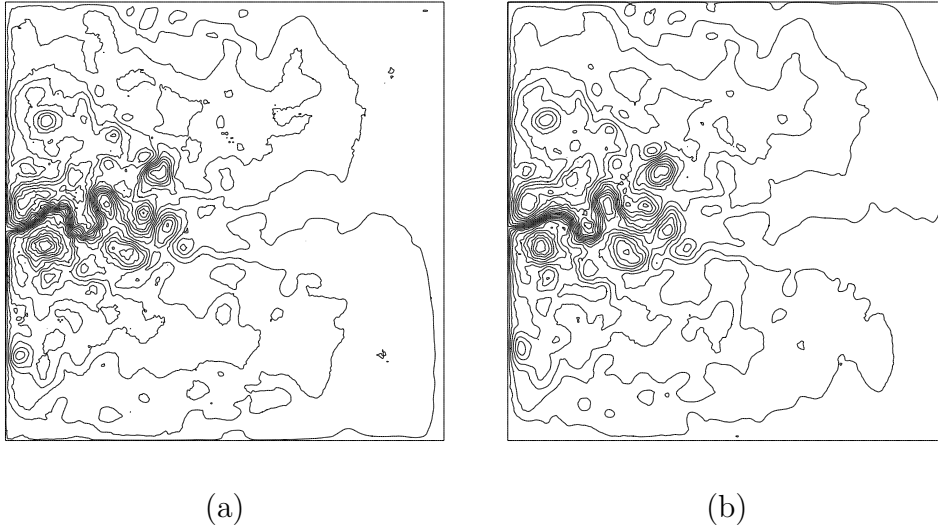


FIG. 6.9 – Résultat de la minimisation de la fonction coût primale avec une erreur modèle de 2%. Solution au début (a) et à la fin (b) de la période d'assimilation.

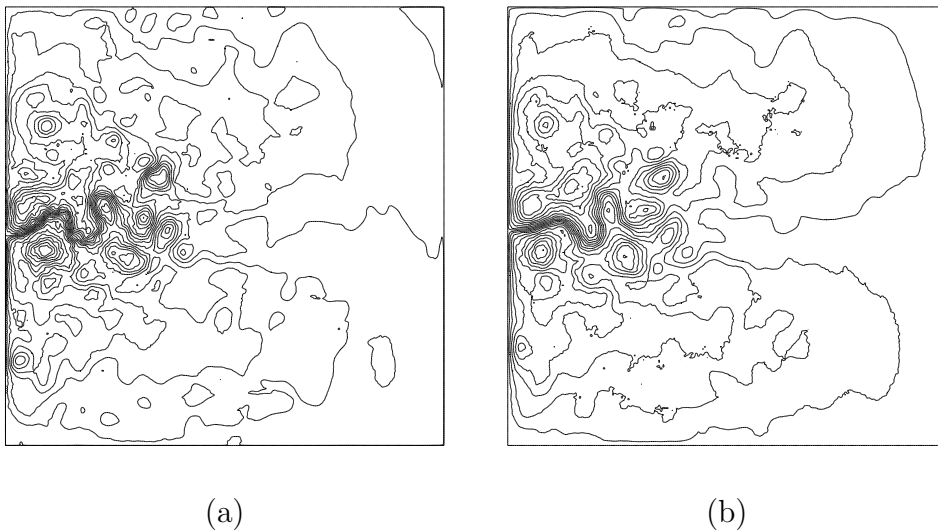


FIG. 6.10 – Résultat de la minimisation de la fonction coût primale avec une erreur modèle de 5%. Solution au début (a) et à la fin (b) de la période d'assimilation.

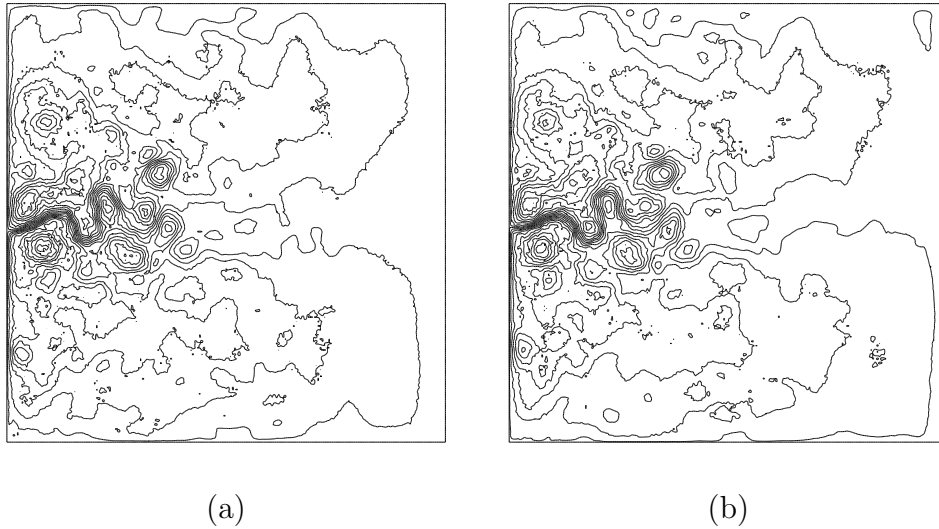


FIG. 6.11 – Résultat de la minimisation de la fonction coût duale avec une erreur modèle de 2%. Solution au début (a) et à la fin (b) de la période d'assimilation.

les lignes de niveau restent relativement lisses, ce qui permet à la solution de pouvoir évoluer en temps pendant de grandes périodes sans devenir totalement bruitée.

La comparaison entre les figures 6.9 et 6.10 montre bien la vitesse à laquelle se dégrade la solution construite par la méthode primale lorsqu'on augmente l'erreur modèle. Il est donc clair que cette méthode est fortement pénalisée par la présence d'un terme d'erreur dans le modèle, même petit.

6.5.2 Méthode duale

Nous allons maintenant appliquer l'algorithme dual pour assimiler les données bruitées lorsqu'une erreur modèle est présente. Nous avons repris la même erreur modèle simulée que dans la section précédente concernant la méthode primale.

Les figures 6.11 et 6.12 (à comparer respectivement aux figures 6.9 et 6.10 montrant les résultats de l'approche primale) montrent les résultats de l'algorithme dual, au début et à la fin de la période d'assimilation, correspondant à une erreur modèle de 2 et 5% respectivement. Nous constatons cette fois que les champs sont assez bruités, mais c'était déjà le cas lorsqu'il n'y avait pas d'erreur modèle. Ce n'est donc pas très surprenant.

Par contre, il semble que la présence de l'erreur modèle influe beaucoup moins sur la reconstitution de l'état initial que dans le cas primal. Cela était également prévisible puisque cette méthode prend directement en compte l'erreur modèle. L'augmentation du terme d'erreur modèle conduit simplement à une petite détérioration de la reconstitution de l'état initial, ainsi que l'augmentation relative des irrégularités sur les lignes de champ.

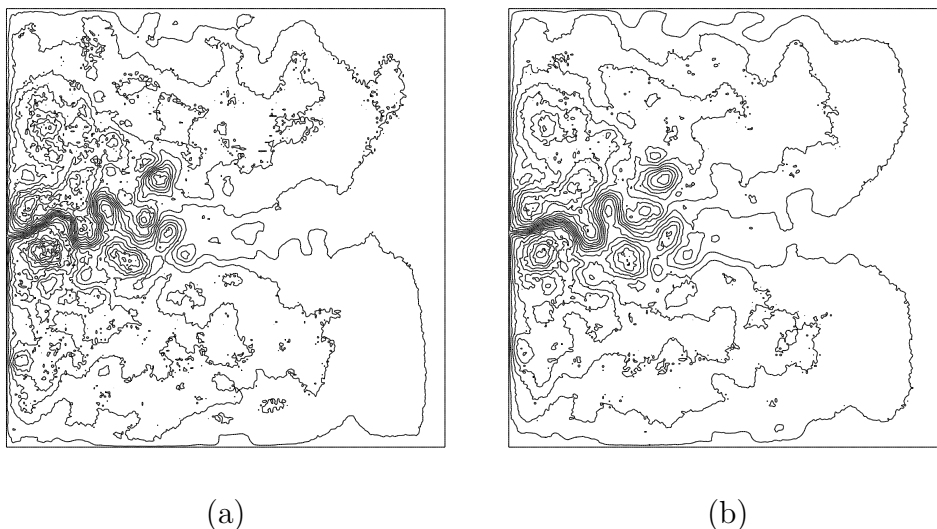


FIG. 6.12 – Résultat de la minimisation de la fonction coût duale avec une erreur modèle de 5%. Solution au début (a) et à la fin (b) de la période d’assimilation.

6.5.3 Comparaison

La figure 6.13 montre les normes RMS des erreurs d’assimilation commises par les méthodes primale et duale en présence d’un terme d’erreur, allant de 0 à 5%, introduit dans le modèle au cours de l’assimilation. En trait plein, on voit l’erreur correspondant à la méthode primale, et en pointillés celle de la méthode duale.

Il apparaît que, comme nous pouvions nous y attendre et comme nous l’avons déjà constaté sur les lignes de champ des solutions identifiées, la méthode duale est beaucoup moins sensible à la présence d’un terme d’erreur dans le modèle que la méthode primale. Il faut tout de même se méfier si l’on augmente trop l’erreur modèle, car il se peut que les équations directes du modèle perturbé divergent très rapidement, le terme censé modéliser les erreurs modèles n’ayant pas le temps de rattraper la croissance des perturbations ajoutées dans le modèle.

La détérioration de la solution primale avec l’erreur modèle est assez claire, et était tout à fait prévisible, puisque l’algorithme primal ne peut pas tenir compte numériquement d’une quelconque erreur dans le modèle. La présence de celle-ci ne fait donc que perturber toutes les trajectoires construites au cours de la minimisation de la fonction coût primale.

L’algorithme dual apparaît donc ici comme la meilleure alternative possible lorsqu’on veut essayer de modéliser l’erreur modèle. En effet, lorsque le terme d’erreur n’est qu’à 2% du signal (ce qui est déjà relativement important tout de même), l’erreur d’assimilation commise par l’approche primale est presque 2 fois plus grande que celle obtenue par l’approche duale. L’écart est encore plus grand lorsque le modèle est biaisé à hauteur de 5%.

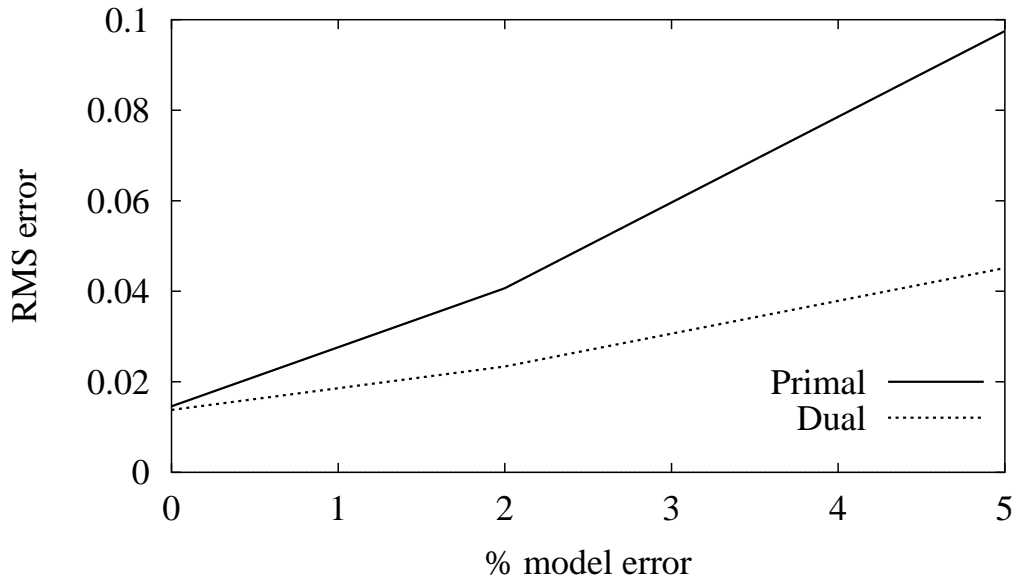


FIG. 6.13 – Normes RMS des erreurs d’assimilation obtenues pour les deux méthodes en fonction de l’erreur introduite dans le modèle au cours de la période d’assimilation.

6.6 Sensibilité aux observations

Nous allons maintenant étudier la sensibilité des deux méthodes au nombre d’observations disponibles. Jusqu’à présent, nous disposions d’observations tous les 5 pas de temps et tous les 5 points de grille en surface. Nous allons réduire ou augmenter leur nombre. Nous avons fait varier la disponibilité des observations, de tous les pas de temps et tous les pas d’espaces sur la grille de surface à tous les 20 pas de temps et tous les 20 pas d’espace en surface. La dimension de l’espace des observations variera ainsi entre 605 et 3272481 (soit 27 fois plus que la dimension de l’espace des états). Nous allons voir les résultats détaillés des minimisations pour ces deux valeurs extrêmes de la dimension de l’espace des observations avant de comparer globalement les deux méthodes.

6.6.1 Méthode primale

La figure 6.14 montre la couche de surface de l’état initial reconstitué et l’état final correspondant de la période d’assimilation, obtenus avec la méthode primale en utilisant 605 observations, c’est-à-dire une tous les 20 pas d’espace, tous les 20 pas de temps.

Les résultats équivalents à la figure précédente sont représentés sur la figure 6.15 lorsqu’on utilise 3272481 observations (tous les pas d’espace sur la grille de surface

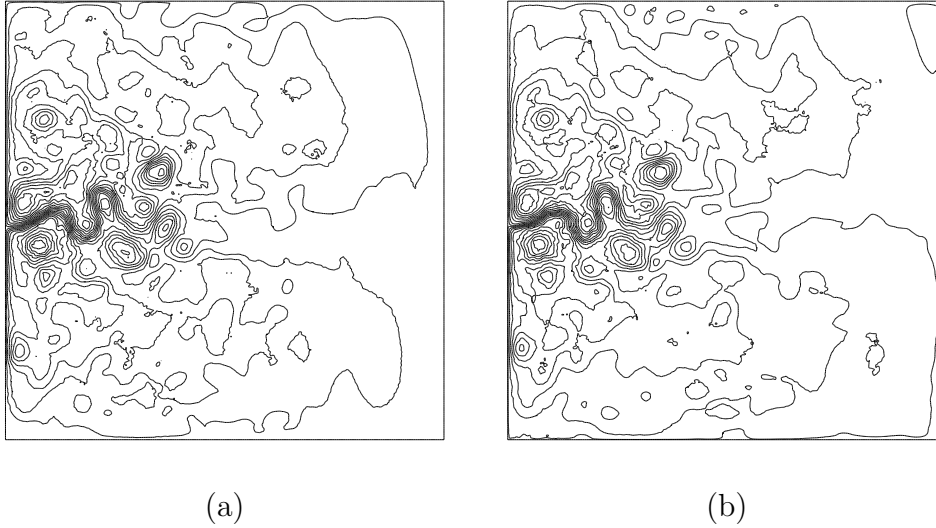


FIG. 6.14 – Résultat de la minimisation de la fonction coût primaire en utilisant 605 observations. Couche de surface de la solution au début (a) et à la fin (b) de la période d'assimilation.

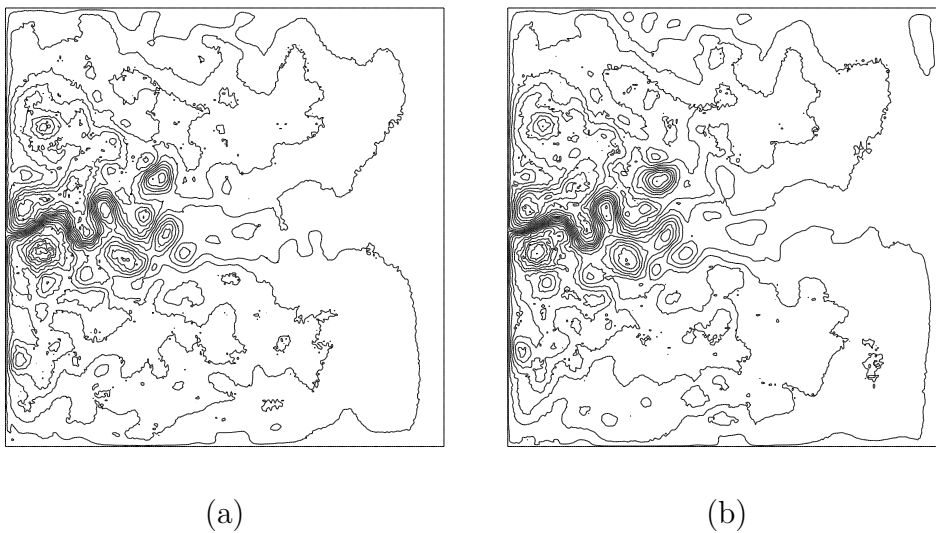


FIG. 6.15 – Résultat de la minimisation de la fonction coût primaire en utilisant 3272481 observations. Couche de surface de la solution au début (a) et à la fin (b) de la période d'assimilation.

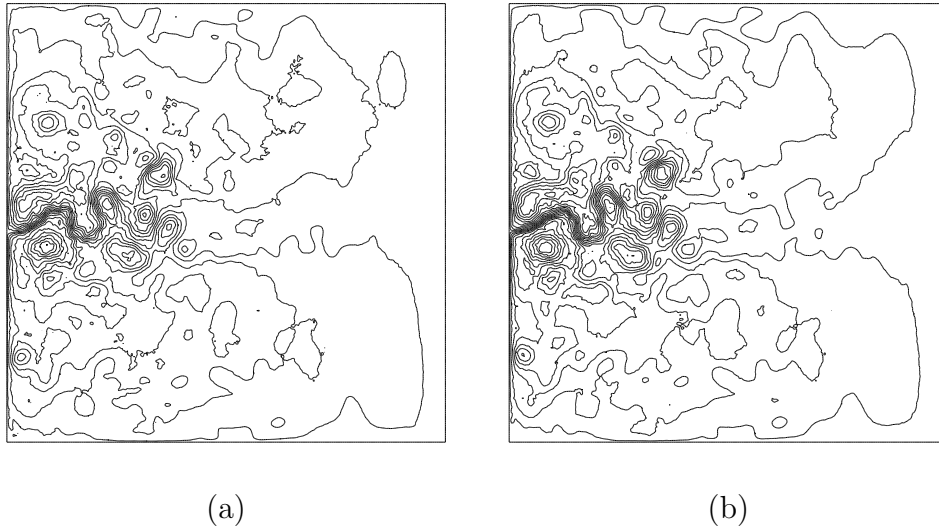


FIG. 6.16 – Résultat de la minimisation de la fonction coût duale en utilisant 605 observations. Couche de surface de la solution au début (a) et à la fin (b) de la période d’assimilation.

et à tous les pas de temps).

On constate que la reconstitution de l’état initial est bien meilleure lorsque beaucoup d’observations sont utilisées, même si les champs obtenus ne sont pas tout à fait lisses. En effet, à tout instant, on souhaite que la trajectoire identifiée soit proche des observations bruitées (dont les champs sont représentés sur la figure 6.3), ce qui laisse peu de place au modèle pour essayer de lisser les champs. Par contre, lorsque peu d’observations sont utilisées, la reconstruction de l’état initial est partielle, mais celui-ci est plus régulier puisque le point de départ de la minimisation est un champ lisse (l’état réel du système mais à un instant différent) et peu de contraintes bruitées agissent sur l’évolution de l’état identifié au cours de la minimisation de la fonction coût.

La figure 6.6, qui montrait les mêmes résultats en utilisant 28577 observations, est une sorte de moyenne entre les deux précédentes figures. En effet, avec des observations tous les 5 pas d’espace, tous les 5 pas de temps, on obtenait une meilleure reconstitution mais un champ moins lisse que lorsque peu d’observations sont utilisées, et une moins bonne reconstitution mais plus lisse qu’avec beaucoup d’observations.

6.6.2 Méthode duale

Les figures 6.16 et 6.17 montrent les résultats des minimisations des fonctions coût duales lorsqu’on utilise respectivement peu (605) et beaucoup (3272481) d’observations. On constate une fois encore que l’augmentation du nombre d’observations améliore l’identification de l’état initial et diminue le lissage des champs. Néanmoins,

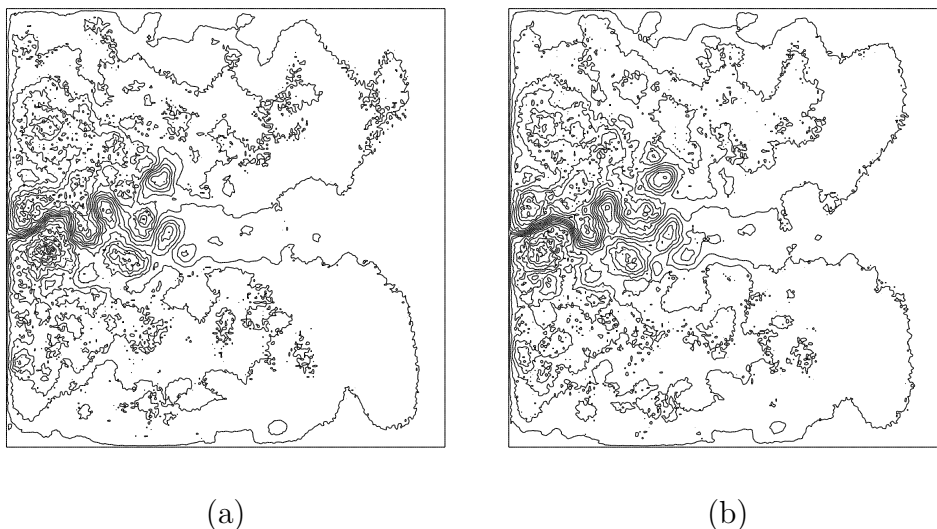


FIG. 6.17 – Résultat de la minimisation de la fonction coût duale en utilisant 3272481 observations. Couche de surface de la solution au début (a) et à la fin (b) de la période d’assimilation.

cette tendance semble plus faible que dans le cas de la méthode primale. De plus, les lignes de niveau sont globalement moins lisses que dans l’approche primale. Ceci avait déjà été constaté dans les premières sections de ce chapitre.

6.6.3 Comparaison

La figure 6.18 montre la norme RMS, moyennée sur toute la période d’assimilation, de l’erreur entre la solution identifiée et la trajectoire de référence en fonction du nombre d’observations disponibles. Celui varie de 605 à 3272481 en agissant sur la fréquence spatiale et temporelle de récupération des observations, qui varie elle de 1 à 20. La courbe des erreurs commises avec la méthode primale est en trait plein et celle de la méthode duale en pointillés.

On constate plusieurs choses. Tout d’abord, lorsque le nombre d’observations disponibles est faible et a tendance à augmenter, l’erreur a tendance à diminuer. En effet, plus on utilise d’observations et meilleure est l’assimilation de données. Néanmoins, à partir d’un grand nombre d’observations (plus que la dimension de l’espace des états), l’erreur a tendance à réaugmenter. Ceci provient essentiellement du fait que le système devient sur-déterminé (plus de données que d’inconnues), les données n’étant pas compatibles entre elles à cause notamment du bruitage.

Si on compare les deux méthodes, on s’aperçoit que la méthode duale est meilleure lorsque le nombre d’observations est inférieur à 100000, soit environ la dimension de l’espace des états. De plus, elle est un peu moins sensible à une diminution de la quantité d’informations. Ceci vient du fait que le nombre d’observations dispo-

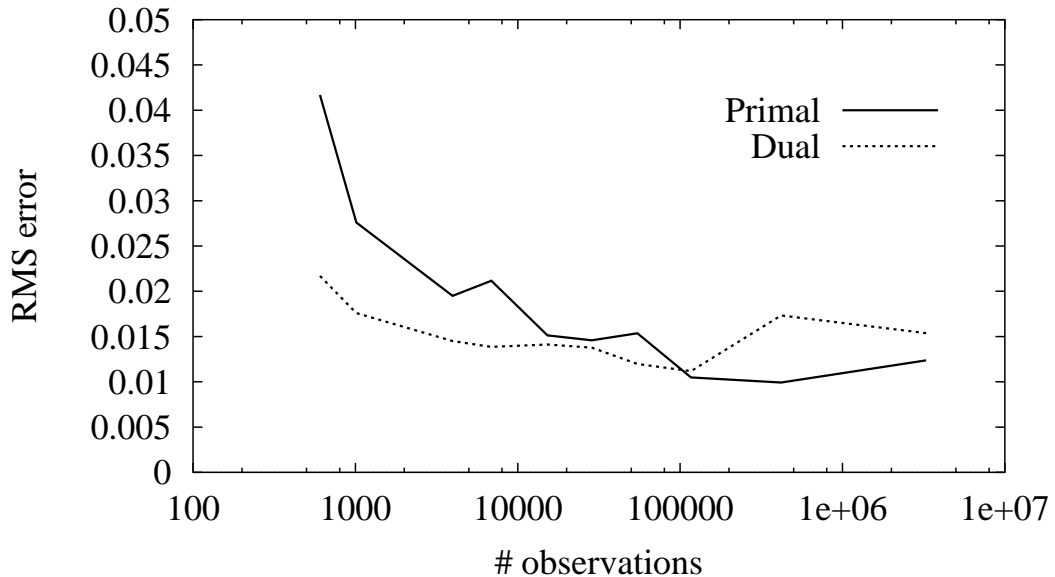


FIG. 6.18 – Normes RMS des erreurs d’assimilation obtenues pour les deux méthodes en fonction de la dimension de l’espace des observations.

nibles pour l’assimilation correspond à la dimension de l’espace de contrôle dual, et la minimisation de la fonction coût est alors beaucoup plus facile et poussée (pour un nombre d’itérations fixé dans l’algorithme de minimisation) ce qui conduit à de meilleurs résultats que la méthode primale. Cette dernière ne bénéficie que très peu de la diminution du nombre d’observations.

Lorsque le nombre d’observations est supérieur à 10^5 , la méthode primale devient meilleure, ou plutôt la méthode duale devient un peu plus mauvaise. Ceci s’explique par un argument symétrique de celui que nous venons de voir pour un petit nombre d’observations. Ici, la dimension de l’espace de contrôle dual est supérieur à celle de l’espace primal. La minimisation est alors plus lente et délicate. De plus, le système devient surdéterminé, et ceci pose plus de problèmes à l’algorithme dual qu’au primal. En effet, la méthode duale est sous-déterminée puisque son vecteur de contrôle est de taille supérieure au nombre d’équations du modèle discrétisé, alors que la méthode primale est plutôt sur-déterminée (plus d’observations que d’inconnues) mais les données n’interviennent que dans la fonction coût et la sur-détermination du problème n’est pas vraiment gênante. Cela conduit juste à une fonctionnelle plus compliquée et donc un peu plus lente à minimiser a priori.

Il semble donc que l’algorithme dual soit légèrement moins sensible au nombre d’observations, et qu’il soit plus performant lorsque la dimension de l’espace des observations est plus petite que celle de l’espace des états. L’écart est assez flagrant lorsque le nombre d’observations utilisées pour l’assimilation est vraiment faible de

vant la dimension de l'espace des états puisque l'erreur d'assimilation de l'algorithme dual est jusqu'à environ deux fois plus faible que l'erreur commise avec la méthode primale.

6.7 Détection de l'erreur modèle

Dans cette section, nous allons faire des expériences jumelles dont le but est d'identifier la condition initiale à partir d'observations générées par un modèle bruité mais sans erreurs d'observation. Nous gardons la même condition initiale exacte (à identifier) que dans les paragraphes précédents, et nous allons faire tourner un modèle direct bruité avec une erreur modèle de 2%. Nous allons récupérer à intervalles de temps et d'espace réguliers (tous les 5 pas d'espace sur la grille et tous les 5 pas de temps) les valeurs de la fonction de courant de surface. Les données ainsi récupérées vont servir d'observations, sans les bruitez (i.e. sans erreur d'observation). Nous allons désormais *oublier* le terme d'erreur dans le modèle et essayer de reconstituer la condition initiale à partir de ces observations. En supposant qu'il n'y a pas d'erreur d'observations, cela revient à tester si les méthodes variationnelles que nous testons sont capables d'identifier la solution d'un problème biaisé par une erreur modèle inconnue. Comme précédemment, les matrices de covariance d'erreur sont cohérentes avec les erreurs introduites.

6.7.1 Condition initiale exacte et observations exactes d'un modèle biaisé

La figure 6.19-a montre la solution exacte en surface à l'instant initial (haut) et à l'instant final (bas). Il s'agit de la même trajectoire que celle présentée en figure 6.1. La figure 6.19-b montre les observations récupérées sans erreur de mesure à partir d'un modèle biaisé à hauteur de 2%, initialisé par la condition initiale exacte. L'observation récupérée à l'instant initial (haut de la figure 6.19-b) est donc exactement l'état initial exact du système (à ceci près que l'observation n'est réalisée que tous les 5 points de grille en espace). Par contre, le simple fait de perturber le modèle engendre une erreur assez importante sur l'état final du système (bas de la figure 6.19-b).

6.7.2 Solutions identifiées par les deux méthodes

À partir de ce jeu d'observations, nous avons essayé de reconstruire numériquement la condition initiale en utilisant les deux méthodes variationnelles (primale et duale).

La figure 6.20 montre (uniquement sur la couche de surface) l'état initial identifié par la méthode primale (a) et par la méthode duale (b). Il apparaît assez clairement que la méthode duale a conduit à une meilleure reconstruction de l'état initial exact.

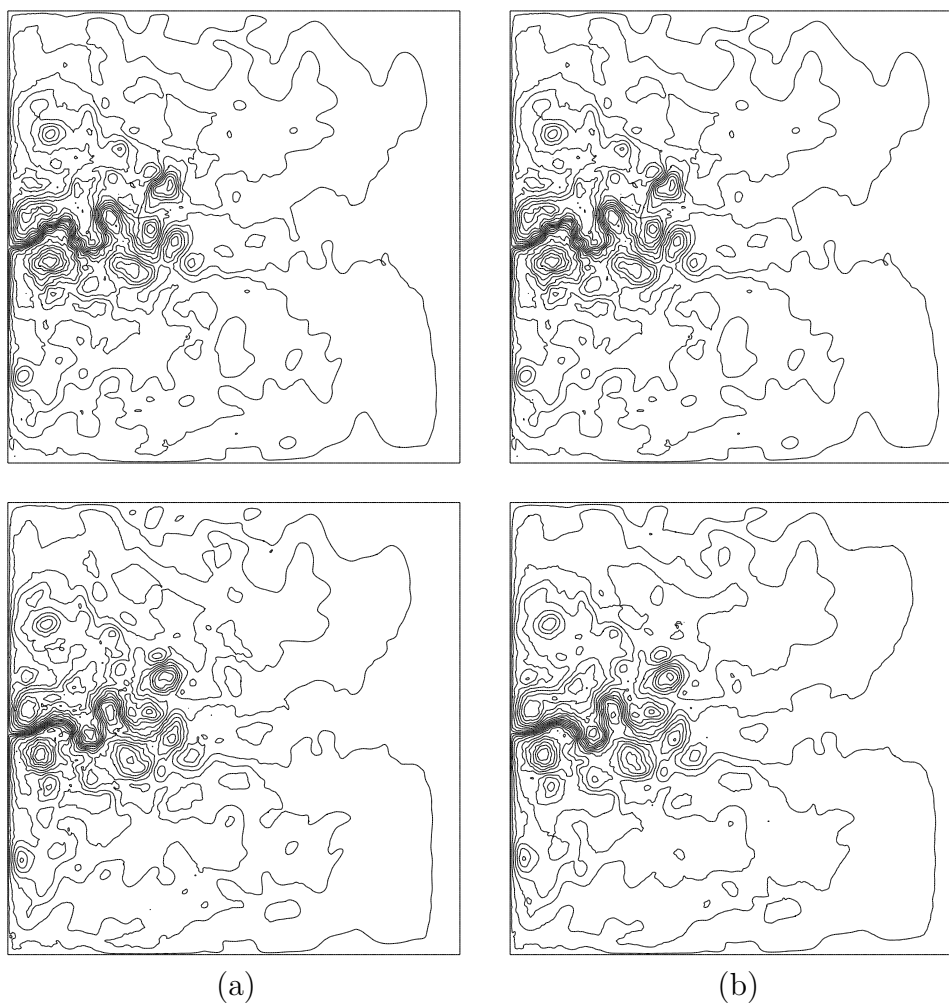


FIG. 6.19 – Solution exacte (sans erreur modèle) à l'instant initial et à l'instant final pour la couche de surface (a). Observations récupérées sans erreur d'observation à partir d'un modèle biaisé à l'instant initial et à l'instant final (b).

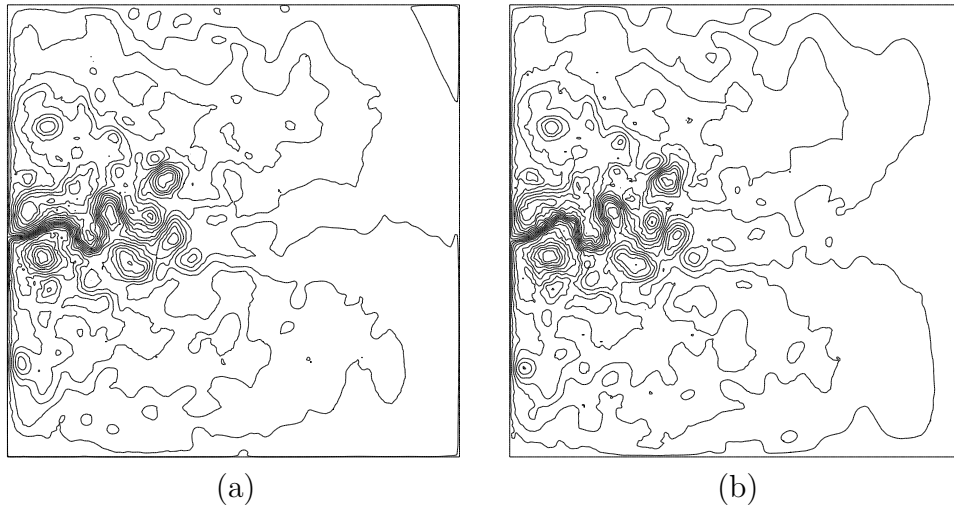


FIG. 6.20 – Condition initiale identifiée par la méthode primale (a) et par la méthode duale (b), sur la couche de surface.

6.7.3 Comparaison des deux méthodes

La figure 6.21 montre pour chacune des deux méthodes la norme RMS de la différence sur la couche de surface entre la trajectoire à reconstituer (issue du modèle bruité) et la trajectoire issue de la condition initiale identifiée. On a également représenté sur cette figure la norme RMS de la différence entre la trajectoire à reconstituer et la trajectoire que l'on obtient en ne cherchant pas à identifier le terme d'erreur modèle, afin de montrer qu'en supposant qu'il y a dans la réalité un terme inconnu dans le modèle, le fait de ne pas en tenir compte peut avoir des conséquences néfastes pour la reconstitution de la trajectoire.

On constate assez évidemment que la différence entre la trajectoire à identifier (provenant du modèle bruité) et celle calculée en ignorant la présence d'un terme d'erreur dans le modèle augmente assez nettement avec le temps. Ceci montre bien que si, dans la *réalité*, le modèle contient des termes d'erreur inconnus, le fait de ne pas en tenir compte dégrade sensiblement la qualité des trajectoires obtenues.

On constate également que les erreurs augmentent avec le pas de temps pour les deux méthodes, mais de façon légèrement moins marquée pour la méthode duale. La trajectoire identifiée par la méthode primale conserve un écart assez régulier avec la trajectoire issue de la condition initiale exacte mais sans tenir compte d'un terme d'erreur dans le modèle. Le modèle utilisé dans la méthode primale ne tenant pas non plus compte d'un terme d'erreur présent dans la réalité, il est donc tout à fait normal que l'algorithme primal donne de moins bons résultats qu'en intégrant simplement le modèle non bruité à partir de la vraie condition initiale.

Par contre, non seulement la différence entre la trajectoire identifiée par la méthode

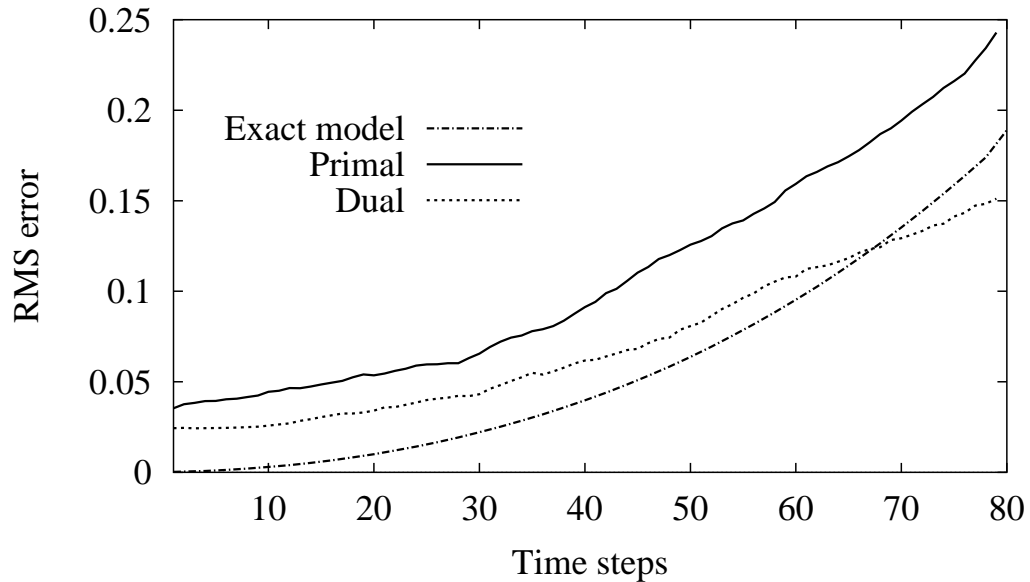


FIG. 6.21 – Normes RMS des erreurs d’assimilation obtenues pour les deux méthodes en fonction du nombre de pas de temps le long de la période d’assimilation.

duale et la trajectoire issue du modèle réel bruité augmente moins vite que pour la méthode primale, mais en plus, à la fin de la période d’assimilation, celle-ci est moins élevée que la différence entre la trajectoire exacte du modèle non bruité et la trajectoire exacte du modèle réel. En effet, la méthode duale construit un terme correctif dans le modèle à partir des données, dans le seul but de reconstruire une trajectoire plus proche de celle qui a conduit à la récupération des observations, et on constate bien sur la figure 6.21 que la trajectoire identifiée par l’algorithme dual s’éloigne moins vite de la réalité que la trajectoire issue de la condition initiale exacte avec un modèle sans terme correctif.

6.8 Conclusions et remarques sur les deux méthodes numériques

Nous avons tout d’abord testé la convergence numérique de l’algorithme dual et nous avons constaté que la fonctionnelle duale était relativement bien diminuée au cours des premières dizaines d’itérations dans l’algorithme de minimisation. Nous avons ensuite cherché à estimer la part d’erreur numérique et la part de non équivalence mathématique entre les deux méthodes sur un problème de plus petite dimension, en les testant tout d’abord sur une version strictement linéaire du modèle, puis sur le modèle non linéaire. Nous avons constaté que la méthode duale fournissait de

meilleurs résultats sur le modèle non linéaire, bien qu'un peu moins bonne sur le modèle linéarisé.

Nous avons ensuite comparé les deux approches dans plusieurs cas de figure. Tout d'abord dans un cadre standard, sans erreur modèle et avec des observations disponibles en quantité relative (tous les 5 points de grille en surface, tous les 5 pas de temps). Nous avons alors constaté que les deux méthodes sont relativement équivalentes, avec un léger avantage numérique pour l'approche duale puisque celle-ci donne de légèrement meilleurs résultats, globalement sur toute la période d'assimilation, que la méthode primale. Néanmoins, les lignes de niveau sont nettement moins lisses, et des prévisions à long terme (si elles pouvaient s'avérer fiables) seraient complètement perturbées par ces irrégularités qui croissent avec le temps.

Lorsque nous rajoutons un terme d'erreur dans le modèle pour simuler toutes les imperfections du modèle (paramètres mal estimés, simplifications inadaptées, ...), la méthode primale montre clairement ses limites alors que la méthode duale est relativement moins perturbée par la présence de ce terme d'erreur. Par contre, la méthode primale conserve des lignes de niveau assez lisses, contrairement aux solutions duales qui ont toujours beaucoup d'irrégularités, qui pourraient s'avérer gênantes pour des prévisions à long terme. Nous avons également constaté la tendance chaotique du modèle puisqu'il a été impossible de mettre un terme d'erreur trop important, sous peine de voir une bonne partie des intégrations directes du modèle exploser numériquement.

Nous avons ensuite testé la sensibilité des deux méthodes à la quantité d'observations disponibles pour l'assimilation. Lorsque le nombre d'observations disponibles est trop faible, les deux méthodes souffrent d'un manque d'information. Lorsque ce nombre est trop élevé au contraire, les minimisations s'avèrent plus délicates et ont tendance à ralentir ou détériorer les reconstitutions de l'état initial. La méthode duale s'avère plus performante lorsque le nombre d'observations est plus petit que la dimension d'un vecteur d'état, puisque elle travaille sur un espace de dimension plus petite que l'algorithme primal, mais lorsqu'on dispose de plus d'observations que de points dans la grille de discrétisation, la méthode duale est pénalisée en travaillant sur un espace trop grand.

Enfin, nous avons constaté que la méthode duale était capable, dans une certaine mesure, de mieux identifier la condition initiale à partir d'observations issues d'un modèle bruité, mais aussi de mieux faire évoluer la trajectoire dans le temps en intégrant un terme correctif dans le modèle et donc de mieux simuler le modèle bruité. La méthode primale n'a pu, quant à elle, que chercher une trajectoire pour le modèle non bruité, et a donc conduit à une identification moins performante.

Chapitre 7

Conclusion

Nous nous sommes intéressés à différents aspects et différentes techniques liés à l'assimilation de données. Nous avons tout d'abord regardé l'aspect minimisation des fonctionnelles. En effet, cela reste à ce jour un problème compliqué puisque les fonctionnelles que l'on rencontre en assimilation de données ne sont que très rarement quadratiques, et vivent dans un espace de très grande dimension. L'évaluation même de la fonction coût est déjà numériquement onéreuse. Nous avons donc testé différentes approches d'une classe d'algorithmes de minimisation, les algorithmes de type BFGS, afin de trouver celle qui était la plus performante pour des problèmes proches de ce qui nous intéresse. Nous avons ainsi constaté qu'il existe un grand choix dans tous les paramètres de ces algorithmes (formules de mise à jour, paramètres d'initialisation) et nous avons introduit une nouvelle façon de mettre à jour les matrices de préconditionnement permettant de gagner en efficacité, et donc en terme de temps de calcul.

Ensuite, dans l'idée un peu simpliste de reconstituer l'état initial à partir notamment d'une condition finale, nous avons regardé deux méthodes de résolution rétrograde qui consistent à intégrer les équations d'un modèle en partant de la condition finale dans l'idée d'identifier la condition initiale. Nous n'avons malheureusement rien pu obtenir de satisfaisant avec la méthode de quasi-réversibilité, qui a montré ses limites sur un problème relativement simple (l'équation de la chaleur rétrograde).

Par contre, le nudging rétrograde a montré qu'il est possible d'obtenir très rapidement une bonne évaluation de l'état initial, permettant éventuellement de servir d'ébauche pour un autre algorithme. En effet, sur le modèle quasi-géostrophique barotrope, nous avons réussi à réduire de façon non négligeable l'erreur sur la condition initiale en seulement une intégration (rétrograde) du modèle. Néanmoins, sous cette forme, le nudging ne permet pas d'obtenir plus qu'une ébauche de la condition initiale et ne permet pas de remplacer les méthodes variationnelles classiques comme le 4D-VAR dans une optique prévisionnelle.

Nous avons ensuite introduit l'algorithme 4D-PSAS, ou algorithme dual, dans un

cadre purement linéaire. Cet algorithme se trouve être mathématiquement équivalent à l'algorithme primal qu'est le 4D-VAR puisque les deux fonctions coût (primale et duale) correspondent aux deux écritures possibles d'un problème min-max, et fournissent donc le même minimum.

Nous avons ensuite testé une extension de l'algorithme dual, dérivée du 4D-PSAS pour fonctionner sur un modèle non linéaire, sur le modèle quasi-géostrophique barocline afin de la comparer avec un algorithme variationnel classique, le 4D-VAR. Nous avons mis en évidence les nombreux avantages de la méthode duale par rapport à l'algorithme primal : une meilleure sensibilité au nombre d'observations, même si la méthode primale s'avère plus performante lorsque le nombre d'observations disponibles devient proche de la dimension du vecteur d'état ; une meilleure sensibilité à la présence d'un terme d'erreur dans le modèle ; la prise en compte inhérente d'un terme de bruit dans le modèle (nous avons constaté que la méthode duale permettait de trouver une bonne estimation de la condition initiale à partir d'observations générées sur un modèle biaisé) ; un gain relatif en rapidité,...

Ce travail nous encourage donc à faire plus de tests numériques sur d'autres configurations d'océans, mais aussi sur des périodes d'assimilation plus longues par exemple. Il faudrait aussi bien évidemment étudier un peu plus en détail la robustesse de la méthode duale sur les problèmes non linéaires mais aussi faire des expériences numériques avec une ébauche plus lointaine de la condition initiale exacte. Il y a aussi probablement bien d'autres façons de concevoir un algorithme dual pour un problème non linéaire. Le but de tout ceci reste assez clairement d'obtenir à court ou moyen terme un système d'assimilation de données opérationnel et réaliste, basé sur un algorithme du type 4D-PSAS. Il ne faut pas non plus négliger d'autres voies, pas forcément encore explorées dans un contexte océanique et dérivées d'études portant sur d'autres domaines scientifiques.

Annexes

Annexe 1

Limited-memory BFGS diagonal preconditioners for a data assimilation problem in meteorology.

F. Veersé, D. Auroux and M. Fisher

Optimization and Engineering, 1 (3) : 323-339, 2000.



Limited-Memory BFGS Diagonal Preconditioners for a Data Assimilation Problem in Meteorology

F. VEERSÉ

INRIA Rhône-Alpes, Monbonnot Saint Martin, France

email: fabrice.veerse@imag.fr

D. AUROUX*

Ecole Normale Supérieure, Lyon, France

email: didier.auroux@ens-lyon.fr

M. FISHER

European Centre for Medium-Range Weather Forecasts, Reading, UK

Received December 17, 1999; Revised September 27, 2000

Abstract. This paper uses two simple variational data assimilation problems with the 1D viscous Burgers' equation on a periodic domain to investigate the impact of various diagonal-preconditioner update and scaling strategies, both on the limited-memory BFGS (Broyden, Fletcher, Goldfarb and Shanno) inverse Hessian approximation and on the minimization performance. These simple problems share some characteristics with the large-scale variational data assimilation problems commonly dealt with in meteorology and oceanography.

The update formulae studied are those proposed by Gilbert and Lemaréchal (*Math. Prog.*, vol. 45, pp. 407–435, 1989) and the quasi-Cauchy formula of Zhu et al. (*SIAM J. Optim.*, vol. 9, pp. 1192–1204, 1999). Which information should be used for updating the diagonal preconditioner, the one to be forgotten or the most recent one, is considered first. Then, following the former authors, a scaling of the diagonal preconditioner is introduced for the corresponding formulae in order to improve the minimization performance. The large negative impact of such a scaling on the quality of the L-BFGS inverse Hessian approximation led us to propose an alternate updating and scaling strategy, that provides a good inverse Hessian approximation and gives the best minimization performance for the problems considered. With this approach the quality of the inverse Hessian approximation improves steadily during the minimization process. Moreover, this quality and the L-BFGS minimization performance improves when the amount of stored information is increased.

Keywords: quasi-Newton, limited-memory BFGS, diagonal updating, quasi-Cauchy, large-scale minimization

1. Introduction

Numerical simulations in meteorology and oceanography consist essentially in integrating a discretized version of the system of partial derivative equations (PDEs) modeling the evolution of the atmosphere and/or the ocean. This system of PDEs comprises a dynamical core derived from Navier-Stokes equations using relevant approximations, an equation of state for the fluid of interest, an equation representing the first law of thermodynamics, and physical parameterizations modeling subgrid-scale processes (convection, radiation,

*Corresponding author.

precipitation, turbulence, surface drag, etc.). To perform this numerical integration, one needs to provide an initial state and possibly boundary conditions. It is the purpose of the so-called “data assimilation” discipline to merge the information coming from the observations with that from the physical laws governing the fluid evolution (available under the form of a numerical model), in order to infer the initial and boundary conditions that will lead to the best-quality simulation or prediction. The so-called “variational data assimilation (VDA) method” tries to achieve this using optimal-control techniques (fitting the model trajectory to the observations), weighting both sources of information by their respective error covariances. Because the dimension of the model state vector is usually large (10^5 – 10^9) and the relation between the model variables are complex, it is not possible in practice to handle the covariance matrix of forecast errors used to weight the information coming from the model (10^{10} – 10^{18} scalar components). Instead, the corresponding linear operator is modeled as the composition of operators that can be managed with current parallel computers. As a result the specified error covariance information is almost always climatologic and does not depend on the underlying dynamics. This is a major deficiency of current implementations of VDA. Recently Veersé proposed a method to specify dynamical forecast error covariances using limited-memory quasi-Newton operators (Veersé, 1999, to appear). For such a method to be efficient in practice, it is required that the limited-memory inverse Hessian approximation be of good quality. This motivated a study for assessing this quality using a simple model sharing some characteristics with atmospheric models. Not surprisingly the way the limited-memory BFGS (L-BFGS) diagonal preconditioner is specified became a touchstone in this study. This led us to assess some of the diagonal preconditioners proposed in the literature (Nocedal, 1980; Gilbert and Lemaréchal, 1989; Liu and Nocedal, 1989; Zhu et al., 1999) and to propose some alternatives.

L-BFGS implements a limited-memory version of the Broyden-Fletcher-Goldfarb-Shanno update formula for the inverse Hessian (Broyden, 1969; Dennis and Moré, 1977; Gilbert and Lemaréchal, 1989):

$$\mathbf{H}^+ = U(\mathbf{H}, \mathbf{y}, \mathbf{s}) = \left(\mathbf{I} - \frac{\mathbf{s} \otimes \mathbf{y}}{\langle \mathbf{y}, \mathbf{s} \rangle} \right) \mathbf{H} \left(\mathbf{I} - \frac{\mathbf{y} \otimes \mathbf{s}}{\langle \mathbf{y}, \mathbf{s} \rangle} \right) + \frac{\mathbf{s} \otimes \mathbf{s}}{\langle \mathbf{y}, \mathbf{s} \rangle} \quad (1)$$

where \mathbf{H}^+ is the updated inverse Hessian, $\mathbf{s} = \mathbf{x}^+ - \mathbf{x}$ is the difference between the new iterate and the previous one, and $\mathbf{y} = \mathbf{g}^+ - \mathbf{g}$ is the corresponding gradient increment. Here $\langle \cdot, \cdot \rangle$ is the scalar product with respect to which the gradient is defined and the minimization is to be performed; $\mathbf{u} \otimes \mathbf{v}$ is the linear operator that to a vector \mathbf{d} associate the vector $\langle \mathbf{v}, \mathbf{d} \rangle \mathbf{u}$. In the limited version (Nocedal, 1980; Gilbert and Lemaréchal, 1989; Liu and Nocedal, 1989) aimed at large-scale unconstrained minimization problems, one can afford to store say m couples of vectors (\mathbf{s}, \mathbf{y}) . The above update formula is used for the first m iterations. For the subsequent ones, the following algorithm is used:

$$\begin{aligned} \mathbf{H}_k^0 &= \mathbf{D}_k, \\ \mathbf{H}_k^{i+1} &= U(\mathbf{H}_k^i, \mathbf{y}_{k-m+i}, \mathbf{s}_{k-m+i}), \quad \text{for } 0 \leq i \leq m-1, \\ \mathbf{H}_k &= \mathbf{H}_k^m \end{aligned} \quad (2)$$

The starting matrix \mathbf{D}_k is diagonal and several formulations have been proposed and tested in Gilbert and Lemaréchal (1989) and by Liu and Nocedal (1989). Recently Zhu et al.

(1999) proposed a variant, based on the quasi-Cauchy relation $\langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{s} \rangle$ that the experiments of Gilbert and Lemaréchal (1989) showed it was beneficial to enforce (by rescaling the diagonal matrix \mathbf{D} before updating it). The M1QN3 minimization code from the INRIA MODULOPT library either specifies \mathbf{D}_k as the identity matrix multiplied by the Oren-Spedicato factor $\langle \mathbf{y}_{k-1}, \mathbf{s}_{k-1} \rangle / \langle \mathbf{y}_{k-1}, \mathbf{y}_{k-1} \rangle$, or updates \mathbf{D}_k using a scaled version of a “diagonalized” BFGS formula (Eq. (4.9) in Gilbert and Lemaréchal (1989)). In our study the focus is put on the latter case, as it is usually more efficient.

During the minimization only the multiplication of the approximate inverse Hessian matrix by a given vector is needed and is performed efficiently using a two-loop recursion proposed by Nocedal (1980), and the corresponding matrices are never formed. It is precisely this aspect that makes the limited-memory inverse BFGS algorithm suitable for large-scale VDA in meteorology and oceanography, as the size of the corresponding matrices may typically reach $10^5 \times 10^5$.

The paper is organized as follows. In the next section, the simple VDA problems used in this study are presented. Section 3 details the updating formulae for the diagonal preconditioner \mathbf{D}_k , that are assessed. The following two sections present the results from numerical experiments for a quadratic and a non-quadratic VDA problem respectively. A brief discussion based on the present case study is then proposed.

2. The variational data assimilation case study

4D variational data assimilation (4D-Var) may be expressed as the minimization of a cost function $\mathcal{J}(\mathbf{x})$ which measures the misfit of a model state vector \mathbf{x} to a set of observations \mathbf{y}_o and an a priori (background) estimate \mathbf{x}_b of the true vector \mathbf{x}_t . The cost function may be written as

$$\mathcal{J}_{\mathbf{x}}(\mathbf{x}) = (\mathbf{x} - \mathbf{x}_b)^T \mathbf{B}^{-1} (\mathbf{x} - \mathbf{x}_b) + (\mathcal{H}(\mathbf{x}) - \mathbf{y}_o)^T \mathbf{O}^{-1} (\mathcal{H}(\mathbf{x}) - \mathbf{y}_o). \quad (3)$$

The matrices \mathbf{B} and \mathbf{O} are the covariance matrices for random errors in $(\mathbf{x}_t - \mathbf{x}_b)$ and $(\mathcal{H}(\mathbf{x}_t) - \mathbf{y}_o)$ respectively. \mathcal{H} is an operator which, applied to \mathbf{x} produces an estimate of the observation vector. In 4D-Var, the observations are spread over a period of time from t_0 to $t_0 + T$, while the control vector represents the state of the atmosphere or the ocean at time t_0 . Therefore \mathcal{H} includes an integration of the model to the observation times and an interpolation to the observation locations.

A useful degree of preconditioning is achieved by rewriting the problem in terms of the transformed control vector $\chi = \mathbf{L}^{-1}(\mathbf{x} - \mathbf{x}_b)$, where $\mathbf{B} = \mathbf{L}\mathbf{L}^T$. Since the dimension of the control vector χ is typically much larger than the dimension of the observation vector \mathbf{y}_o , this transformation gives a Hessian matrix which has many eigenvalues equal to one. The transformed problem is then written as

$$\mathcal{J}_{\chi}(\chi) = \chi^T \chi + \delta^T \mathbf{O}^{-1} \delta \quad (4)$$

where

$$\delta = (\mathcal{H}(\mathbf{x}) - \mathbf{y}_o). \quad (5)$$

For the atmosphere and the ocean at basin scales the spatial error correlations of the background (a priori) estimate imply that the observations have little influence on the solution (called *analysis*) at the smallest scales, provided the observations are well separated. For this reason, and to reduce the computational burden of computing the analysis in operational numerical weather prediction, the minimization problem is usefully modified to its incremental version (Courtier et al., 1994):

$$\mathcal{J}_\chi^{\text{inc}}(\chi) = \chi^T \chi + \delta^T \mathbf{O}^{-1} \delta \quad (6)$$

where

$$\delta = (\mathcal{H}(\mathbf{x}_b) + \mathcal{H}'_S(\mathbf{S}\mathbf{x}_b) \cdot (\mathbf{x} - \mathbf{x}_b) - \mathbf{y}_o). \quad (7)$$

Here, $\chi = \mathbf{L}^{-1}\mathbf{S}(\mathbf{x} - \mathbf{x}_b)$ where \mathbf{S} is a simplification operator (e.g. a projection to a lower spatial resolution) and $\mathcal{H}'_S(\mathbf{S}\mathbf{x}_b)$ is the derivative of the simplified equivalent of \mathcal{H} evaluated at the simplified background state, $\mathbf{S}\mathbf{x}_b$.

In this simple case study, the evolution model is the viscous Burgers' equation over a one-dimensional cyclic domain:

$$\frac{\partial x}{\partial t} + \frac{1}{2} \frac{\partial x^2}{\partial s} - \nu \frac{\partial^2 x}{\partial s^2} = 0 \quad (8)$$

where s represents the distance in meters around the 45°N constant-latitude circle. The period of the domain is roughly 28.3×10^6 m. The diffusion coefficient is set to $10^5 \text{ m}^2 \text{ s}^{-1}$, as in the experiments by Fisher and Courtier (1995).

The corresponding numerical model used in the calculation of $\mathcal{H}(\mathbf{x})$ represents \mathbf{x} as a vector of discrete Fourier coefficients. The nonlinear term is calculated using the transform method on a non-aliasing grid. A conditionally stable (Matsuno) predictor-corrector scheme is used for the time integration.

For each element of the vector of observations, \mathbf{y}_o , the corresponding element of $\mathcal{H}(\mathbf{x})$ (resp. $\mathcal{H}'_S(\mathbf{S}\mathbf{x}_b) \cdot (\mathbf{x} - \mathbf{x}_b)$) is the result of a model (resp. linearized model) integration followed by an inverse Fourier transform, followed by linear interpolation from the two grid points closest to the observation.

The observations are specified at regular time intervals at fixed locations which are chosen randomly, but with a probability distribution resembling the longitudinal density of radiosonde stations between 30°N and 60°N. The observed values are calculated by applying \mathcal{H} to \mathbf{x}_t and then adding a random error. The observation errors are normally distributed, uncorrelated and all observation errors have the same variance. The covariance matrix of observation error, \mathbf{O} , is therefore proportional to the identity matrix.

The specification of the preconditioning operator \mathbf{L} defining the background error covariance matrix \mathbf{B} is chosen to resemble that formerly used operationally at Météo-France and ECMWF (Courtier et al., 1993).

The scalar product $\langle \cdot, \cdot \rangle$ used for the definition of the gradient and to perform the minimization is given by

$$\langle \mathbf{x}_1, \mathbf{x}_2 \rangle = \mathbf{x}_1^T \mathbf{P} \mathbf{x}_2 \quad (9)$$

where \mathbf{P} is a diagonal matrix with diagonal elements equal to 2, except that corresponding to the constant term of the Fourier decomposition, which is set to 1.

3. Update formulae

This section details the four diagonal-preconditioner update formulae studied: three of them are taken from Gilbert and Lemaréchal (1989), and the remaining one is a generalized version of the quasi-Cauchy update formula (Zhu et al., 1999).

Inverse BFGS formula

The inverse BFGS diagonal-preconditioner update formula is given by Eq. (4.6) in Gilbert and Lemaréchal (1989). The i -th updated diagonal component is:

$$D_+^{(i)} = D^{(i)} + \left(\frac{1}{\langle \mathbf{y}, \mathbf{s} \rangle} + \frac{\langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle}{\langle \mathbf{y}, \mathbf{s} \rangle^2} \right) \langle \mathbf{s}, \mathbf{e}_i \rangle^2 - \frac{2D^{(i)} \langle \mathbf{y}, \mathbf{e}_i \rangle \langle \mathbf{s}, \mathbf{e}_i \rangle}{\langle \mathbf{y}, \mathbf{s} \rangle} \quad (10)$$

It is obtained as the i -th diagonal component of the matrix resulting from updating \mathbf{D} with the inverse BFGS formula. Here $(\mathbf{e}_i)_{1 \leq i \leq n}$ is an orthonormal basis of \mathbb{R}^n for the scalar product $\langle \cdot, \cdot \rangle$.

Direct BFGS formula

The direct BFGS diagonal-preconditioner update formula corresponds to Eq. (4.7) in Gilbert and Lemaréchal (1989):

$$D_+^{(i)} = \left(\frac{1}{D^{(i)}} + \frac{\langle \mathbf{y}, \mathbf{e}_i \rangle^2}{\langle \mathbf{y}, \mathbf{s} \rangle} - \frac{(\langle \mathbf{s}, \mathbf{e}_i \rangle / D^{(i)})^2}{\langle \mathbf{D}^{-1}\mathbf{s}, \mathbf{s} \rangle} \right)^{-1} \quad (11)$$

It results from taking the inverse of the diagonal of the matrix obtained by updating \mathbf{D}^{-1} with the direct BFGS formula.

Inverse DFP formula

The inverse DFP diagonal-preconditioner update formula is given by Eq. (4.8) in Gilbert and Lemaréchal (1989):

$$D_+^{(i)} = D^{(i)} + \frac{\langle \mathbf{s}, \mathbf{e}_i \rangle^2}{\langle \mathbf{y}, \mathbf{s} \rangle} - \frac{(D^{(i)} \langle \mathbf{y}, \mathbf{e}_i \rangle)^2}{\langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle} \quad (12)$$

It is the diagonal of the matrix obtained by updating \mathbf{D} with the inverse DFP formula.

Quasi-Cauchy formula

The quasi-Cauchy diagonal-preconditioner update formula is an extension of Eq. (9) in Zhu et al. (1999) to the case of a general metric defined by the scalar product $\langle \cdot, \cdot \rangle$. It may be written as

$$\mathbf{D}_+ = \begin{cases} \mathbf{D} & \text{if } \langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{s} \rangle \\ (\mathbf{I} + \nu\mathbf{G})^{-2}\mathbf{D} & \text{if } \langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle \neq \langle \mathbf{y}, \mathbf{s} \rangle \end{cases} \quad (13)$$

where \mathbf{G} is the diagonal matrix whose i -th diagonal component is $\langle \mathbf{y}, \mathbf{e}_i \rangle^2$, and ν is the largest solution of $F(\nu) = \langle \mathbf{y}, \mathbf{s} \rangle$ with

$$F(\nu) = \langle (\mathbf{I} + \nu\mathbf{G})^{-2}\mathbf{D}\mathbf{y}, \mathbf{y} \rangle. \quad (14)$$

This diagonal-preconditioner update formula is obtained by solving the minimization problem

$$\min \langle \mathbf{w}, \mathbf{w} \rangle \quad \text{such that } \langle (\mathbf{D}^{1/2} + \Omega)^2\mathbf{y}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{s} \rangle \quad (15)$$

where Ω is a diagonal matrix whose nonzero components are given by the corresponding components of the vector \mathbf{w} .

Scaling

As shown in Gilbert and Lemaréchal (1989), a large number of iterations and function/gradient evaluations may be saved by scaling the diagonal matrix \mathbf{D} before updating it, that is multiplying it by $\langle \mathbf{y}, \mathbf{s} \rangle / \langle \mathbf{D}\mathbf{y}, \mathbf{y} \rangle$ so that the diagonal matrix to be updated satisfies the quasi-Cauchy relation.

The impact of this scaling on the quality of the L-BFGS inverse Hessian approximation will be assessed for the first three diagonal-preconditioner update formulae. This is irrelevant for the quasi-Cauchy diagonal-preconditioner update formula (13): scaling the diagonal to be updated would result in using the identity matrix multiplied by the Oren-Spedicato factor at each iteration.

4. The quadratic case

The quality of the L-BFGS inverse Hessian approximation is first studied for the incremental formulation of 4D-Var (Eq. (6)), an unconstrained quadratic minimization problem. The simplification operator consists in using a lower spectral truncation (number of terms retained in the discrete Fourier series). To enable a large number of computations, the dimension of the high-resolution fields \mathbf{x} is taken to be 258. The low-resolution fields $\mathbf{S}\mathbf{x}$ and the control variable have dimension 130. Other parameters are summarized in Table 1. Some values need some comments: ε_m is the machine epsilon, approximatively equal to 2.220×10^{-16} since all the studies are performed using IEEE 64-bit floating-point

Table 1. Parameters for the quadratic problem.

Parameter	Value
Number of (\mathbf{s}, \mathbf{y}) couples	5
Minimum l_∞ distance between successive iterates	ε_m
Wolfe's line-search parameters	$\alpha = 10^{-4}$ $\beta = 0.9$
Max. number of iterations	120
Max. number of function/gradient evaluations	144
Expected decrease at first iteration	$\mathcal{J}_x^{inc}(0)/2$
Minimum expected final/initial gradient norm ratio	$\sqrt{\varepsilon_m}$

arithmetic. The expected decrease at the first iteration is used to obtain an estimate of the step-size at the first iteration, taken equal to 2 times this value divided by the initial-gradient norm squared (defined by the scalar product $\langle \cdot, \cdot \rangle$ mentioned in Section 2). The minimum ratio of the final to the initial gradient norms is the convergence criterion used for the minimization.

The quality of the L-BFGS inverse Hessian approximation is assessed by computing the eigen-spectrum of

$$\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1} \quad (16)$$

where \mathbf{H}_{true}^{-1} is the true Hessian computed using a second-order adjoint method (Wang et al., 1992; Le Dimet et al., 1997), and \mathbf{H}_{L-BFGS}^{-1} is the L-BFGS Hessian approximation built as in Veersé (to appear).

In order to have a relative measure of this quality, the eigen-spectrum of

$$\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS} \quad (17)$$

is also computed. Since \mathbf{H}_{true}^{-1} and \mathbf{H}_{L-BFGS} are symmetric to a high accuracy, the eigen-spectrum of $\mathbf{I} - \mathbf{H}_{L-BFGS} \mathbf{H}_{true}^{-1}$ is almost identical to the latter one, to which we thus restrict our attention.

Both eigen-spectra are computed using an Implicit Restarted Arnoldi method (Lehoucq et al., 1997).

Although we are interested primarily in the quality of the L-BFGS inverse Hessian approximation, the efficiency of the minimization algorithm is also of concern and will be measured by the number of iterations needed to achieve convergence and the corresponding number of simulations (evaluations of the cost function and its gradient).

4.1. The full-memory case

Before studying the limited-memory versions, the convergence of the full-memory BFGS was checked. This was done by using L-BFGS with a value of the storing index m greater

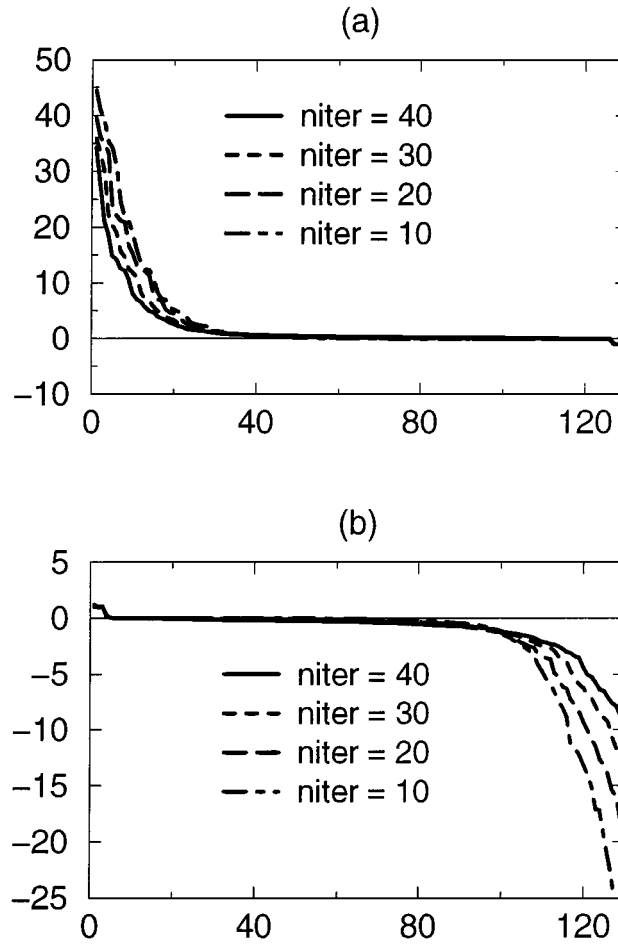


Figure 1. Eigen-spectra of operators for various iteration numbers of the minimization algorithm in the full-memory case. (a) Hessian difference operator $\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$. (b) Relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS}$.

than the total number of iterations required (equal to 40). Figure 1 shows the eigen-spectra of (16) and (17) for various iteration indices; the convergence is evidenced.

4.2. Choice of the (s, y) pair

A degree of freedom is given for the choice of the vectors \mathbf{s} and \mathbf{y} used to update the diagonal preconditioner. The L-BFGS algorithm (Eq. (2)) shows that the diagonal preconditioner should resemble \mathbf{H}_{k-m-1} as much as possible. This suggests to use the pair that is about to be dropped—hereafter called the *oldest pair*— $(\mathbf{s}_{k-m}, \mathbf{y}_{k-m})$ for updating the diagonal matrix \mathbf{D}_k in order to obtain \mathbf{D}_{k+1} . For the first m iterations the initial diagonal matrix (usually the

identity matrix scaled by the Oren-Spedicato factor) is to be used. This is consistent with the fact that all the curvature information from the last m iterations is completely retained in the (\mathbf{s}, \mathbf{y}) couples. However one of us advocated that the diagonal preconditioner should be updated with the newly computed couple of vectors $(\mathbf{s}_k, \mathbf{y}_k)$ —hereafter called *the newest pair*. This puts more weight onto the most recent curvature information, whose quality will improve as the minimization proceeds in the non-quadratic case.

To study the impact of both choices on the quality of the inverse Hessian approximation, the update formulae of Section 3 are implemented without scaling. The corresponding eigen-spectra are shown in figure 2. It is clear that the use of the newest pair provides an inverse Hessian approximation of far better quality than does that of the oldest pair. Moreover the panel (d) suggests that the direct BFGS and quasi-Cauchy update formulae are more accurate than the other two.

Table 2 shows the corresponding number of iterations and simulations (joint evaluations of the function and its gradient) needed to achieve convergence. Except for the direct BFGS update formula, using the newest pair increases the number of simulations, even if

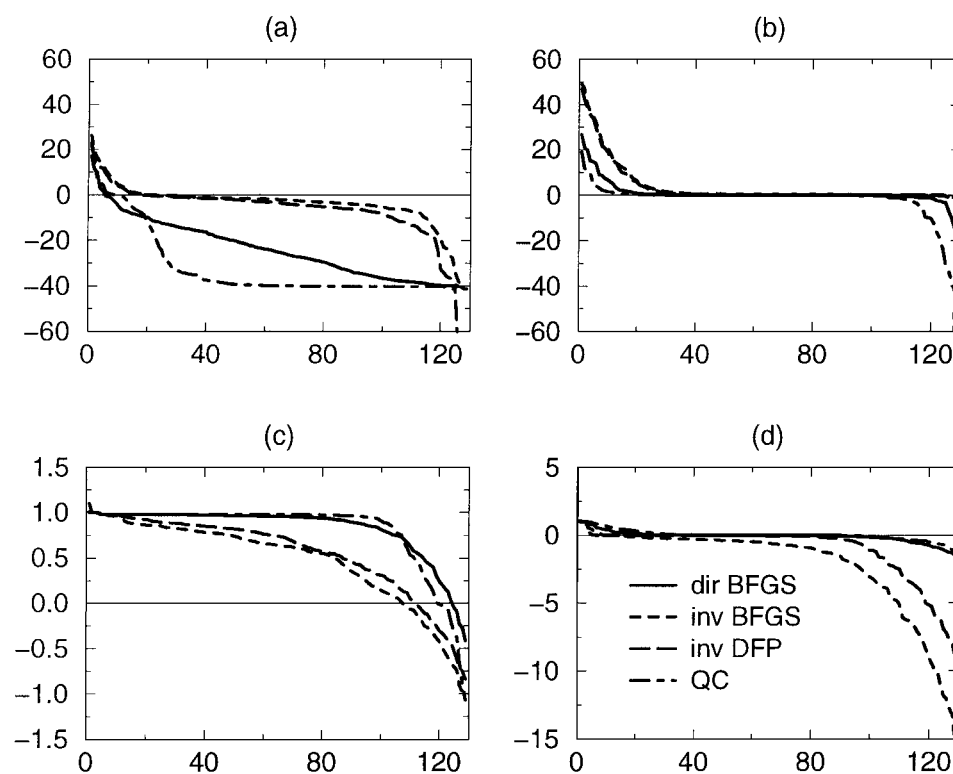


Figure 2. Eigen-spectra of the Hessian difference operator $\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$ for various update formulae using (a) the oldest and (b) newest pair respectively. (c) and (d) show the corresponding eigen-spectra of the relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS}$ for the oldest-pair and newest-pair cases respectively.

Table 2. Performance without scaling, using the oldest or newest pair for updating the diagonal preconditioner (# iterations/# simulations).

Formula	Oldest pair used	Newest pair used
Direct BFGS	77/78	40/52
Inverse BFGS	69/70	42/84
Inverse DFP	68/69	42/84
Quasi-Cauchy	96/103	120/130

it reduces the number of iterations for inverse BFGS and inverse DFP. The quasi-Cauchy update formula has the poorest performance on this quadratic problem. Using the newest pair with direct BFGS is beneficial both for approximating the inverse Hessian and for the minimization.

4.3. Impact of scaling

Gilbert and Lemaréchal (1989) show that scaling the diagonal matrix to force it to satisfy the quasi-Cauchy relation may lead to a better performance of the minimization. The M1QN3 minimization code of the MODULOPT library from INRIA implements the direct BFGS formula using the newest pair and scaling the diagonal before updating it. This leads to the equivalent scaled formula (4.9) in Gilbert and Lemaréchal (1989).

The impact of such a scaling for our case-study problem is now assessed for the first three diagonal-preconditioner update formulae of Section 3. This scaling is irrelevant for the quasi-Cauchy update formula, as already mentioned. Both options of scaling the diagonal matrix before or after updating it are studied. Figure 3 shows the eigenspectra of (16) and (17) for both cases. A comparison with the panels (b) and (d) of figure 2 clearly shows, for both options, a detrimental impact of the scaling onto the quality of the L-BFGS inverse Hessian approximation.

The impact in terms of minimization performance may be assessed from Table 3. Scaling leads to some reduction in the number of function and gradient evaluations needed to converge. Scaling the diagonal preconditioner after updating it is more efficient than doing it before, especially for the inverse BFGS formula. All three methods perform more or less equivalently when the scaling is done after updating, with a slight advantage for the direct BFGS diagonal-preconditioner update formula.

Table 3. Performance using the newest pair, when scaling the diagonal preconditioner before or after updating it (# iterations/# simulations).

Formula	Scaling before updating	Scaling after updating
Direct BFGS	47/49	47/49
Inverse BFGS	55/60	51/53
Inverse DFP	52/53	50/52

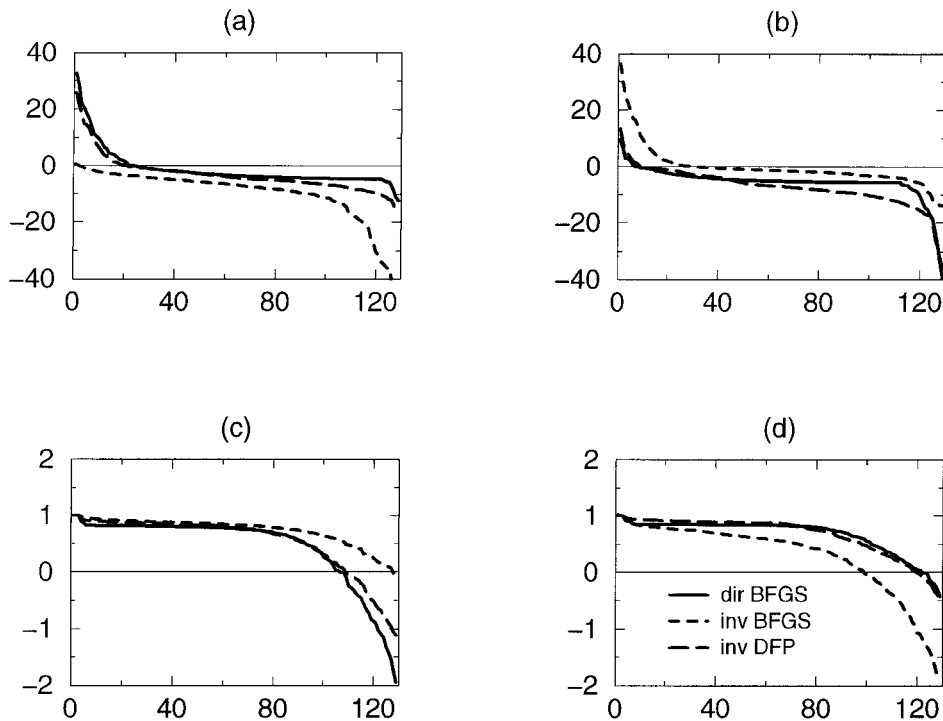


Figure 3. Eigen-spectra of the Hessian difference operator $\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$ when scaling the diagonal preconditioner (a) before updating it and (b) after updating it. (c) and (d) show the eigen-spectra of the relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS}$ corresponding to (a) and (b) respectively.

4.4. A new approach

Let us summarize our findings up to now. To obtain a good approximation of the inverse Hessian the newest pair should be used to update the diagonal preconditioner. But this reduces the minimization performance, except for direct BFGS. It is recommended to scale the diagonal preconditioner after updating, as this always leads to some improvements for the minimization. However the quality of the L-BFGS inverse Hessian approximation is then largely damaged.

It is natural therefore to consider a new approach where a scaled diagonal preconditioner is used for the minimization but the original (unscaled) one is updated. Figure 4 shows the corresponding eigenspectra of (16) and (17). As could be expected, an inverse Hessian quality similar to that of panels (b) and (d) in figure 2 is recovered. The possible differences between both figures may be explained by the different sequences of iterates generated.

Table 4 gives the corresponding impact on the minimization performance, in terms of iterations and simulations. This approach gives a further improvement in terms of simulations required for all three update formulae. All three perform nicely, with a slight advantage for the direct BFGS one.

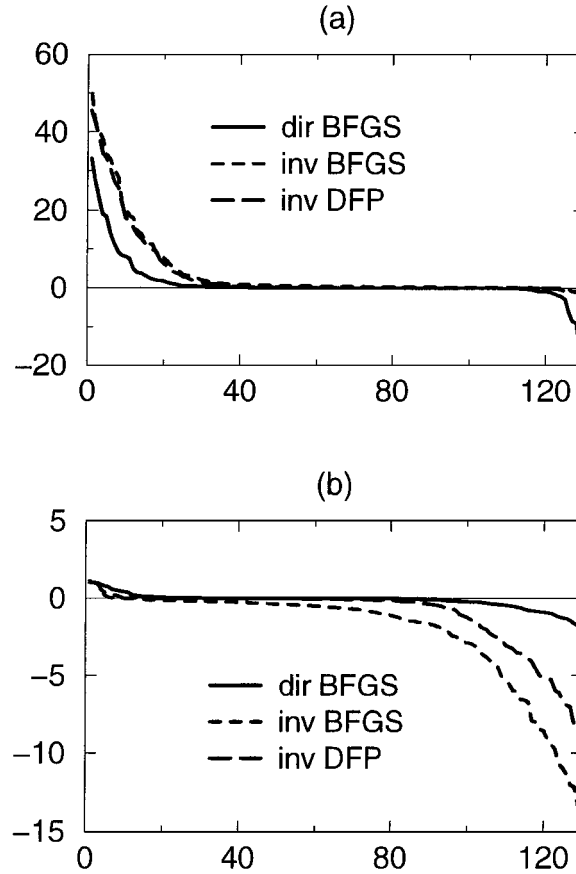


Figure 4. Eigen-spectra of (a) the Hessian difference operator $\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$ and (b) the relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1}\mathbf{H}_{L-BFGS}$ when a scaled diagonal preconditioner is used for the minimization but the unscaled one is updated.

4.5. Impact of m

The focus is now put on how the quality of the L-BFGS inverse Hessian approximation and the minimization performance are affected by a change in the number m of (\mathbf{s}, \mathbf{y}) couples used. This is studied using our best configuration, namely updating the unscaled diagonal preconditioner with direct BFGS using the newest pair but using its scaled version for the minimization. Figure 5 shows the corresponding eigenspectra. There is not much difference on the quality of the L-BFGS inverse Hessian approximation for different values of m . However the approximation tends to improve with increasing values of the storing index.

The corresponding minimization performance are given in Table 5. The performance increases with increasing values of the storing index and the usual optimum value between 3 and 20 is not found with this diagonal-preconditioner update strategy. This is related to the corresponding increase in the quality of the L-BFGS inverse Hessian approximation.

Table 4. Performance using the newest pair, when using a scaled diagonal preconditioner for the minimization but updating the original unscaled one.

Formula	Iterations/Simulations
Direct BFGS	40/43
Inverse BFGS	44/46
Inverse DFP	43/46

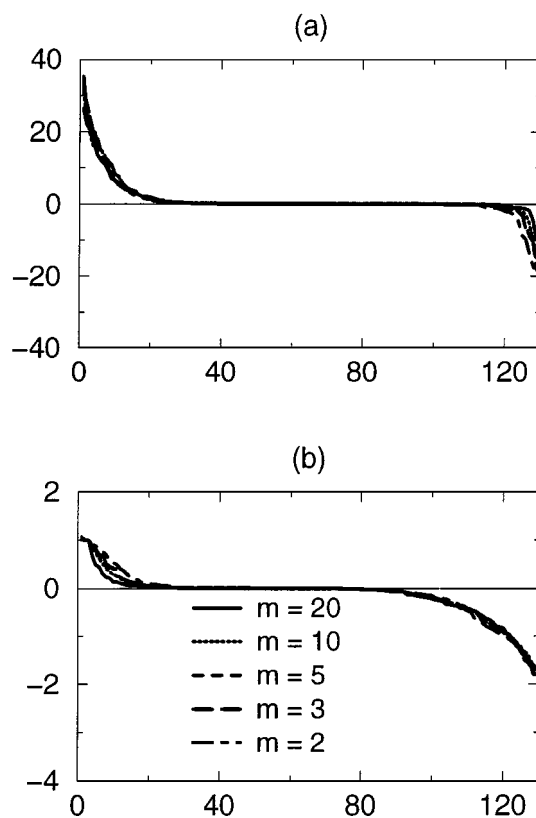


Figure 5. Eigen-spectra of (a) the Hessian difference operator $\mathbf{H}_{true}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$ and (b) the relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS}$ for different values of the storing index m . The diagonal-preconditioner update formula is direct BFGS with the new approach.

4.6. Evolution during the minimization

In variational data assimilation, an approximation of the error covariances of the sought initial condition of the model is provided by the inverse Hessian at the minimum (Veersé, 1999, to appear). However due to the corresponding computational burden for realistic VDA problems in meteorology and oceanography the minimization is usually stopped

Table 5. Minimization performance for various storing indices. The diagonal preconditioner is updated using the direct BFGS formula with the new approach.

m	Iterations/Simulations
2	44/48
3	42/46
5	40/43
10	36/38
20	35/37

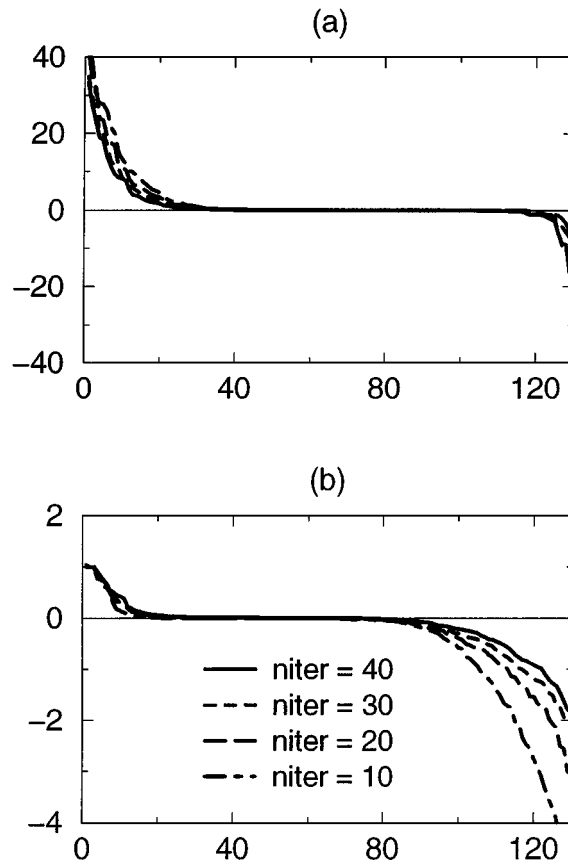


Figure 6. Eigen-spectra of (a) the Hessian difference operator $\mathbf{H}^{-1} - \mathbf{H}_{L-BFGS}^{-1}$ and (b) the relative operator $\mathbf{I} - \mathbf{H}_{true}^{-1} \mathbf{H}_{L-BFGS}$ for different values of the iteration index. The diagonal preconditioner is updated using direct BFGS with the new approach.

before reaching convergence, after a few tens of iterations have been performed. For this reason, it is interesting to see how the quality of the L-BFGS inverse Hessian approximation evolves with increasing iteration indices. Figure 6 shows the evolution of the corresponding eigen-spectra, using our best-case diagonal-preconditioner update formula with $m = 5$ (\mathbf{s}, \mathbf{y}) couples. Clearly, the quality of the approximation improves as the minimization proceeds. This is a natural evolution since the dimension of the subspace explored during the minimization increases whereas the true Hessian is constant for the present quadratic problem.

5. The non-quadratic problem

All the experiments of the previous section have been performed again using the standard formulation of 4D-Var (4), a non-quadratic minimization problem. The dimension of the control variable is 258, identical to the model phase space dimension. Now the Hessian depends on the point where it is evaluated. The quality of the various L-BFGS inverse Hessian approximations was assessed with respect to the Hessians computed with second-order adjoint techniques at the corresponding computed optimal points. The parameters used for the minimization are again those of Table 1, except the maximum numbers of iterations and simulations allowed have been increased to 200 and 250 respectively. The results do not differ qualitatively from those of the quadratic case and the plots of the corresponding eigen-spectra (not shown) are similar to those of the previous section.

A significant difference however is the failure of the first three diagonal-preconditioner update methods when the newest pair is used but no scaling is applied. This occurs at the second iteration during the model integration used for the cost-function and gradient computations. A likely explanation is the generation of an iterate during the minimization that after a few time steps leads to a violation of the stability condition of the model, which then explodes numerically.

The minimization performance may be assessed from Table 6. The standard (non-quadratic) 4D-Var cost function (Eq. (4)) requires a few more iterations and simulations to be minimized than its incremental (quadratic) approximation (Eq. (6)). The main conclusion is the same as for the quadratic case: using the direct BFGS diagonal-preconditioner update formula with the new scaling approach provides both the best performance for the minimization and the best inverse Hessian approximation.

Table 6. Minimization performance for the non-quadratic cost function (# iterations/# simulations).

Formula	No scaling oldest pair	No scaling newest pair	Scaling before newest pair	Scaling after newest pair	New approach newest pair
Direct BFGS	78/79	Failed	52/56	47/49	43/45
Inverse BFGS	76/77	Failed	63/67	69/71	48/52
Inverse DFP	74/75	Failed	55/57	56/58	48/51
Quasi-Cauchy	98/139 ¹	188/248 ²			

6. Discussion

A simple variational data assimilation problem was used as a case-study to assess the impact of various strategies for scaling and updating the L-BFGS diagonal preconditioner, both on the quality of the L-BFGS inverse Hessian approximation and on the minimization performance. The former was evaluated from comparison with the Hessian provided by second-order adjoint techniques, using eigen-decompositions. This approach is feasible only with relatively small-size problems, due to the computational burden of computing these eigen-decompositions.

The minimization performance was measured in terms of number of iterations and simulations required to achieve convergence. Both points of view lead to a few constataions:

- Using the newest (\mathbf{s}, \mathbf{y}) pair to update the diagonal preconditioner gives a better inverse Hessian approximation and, except for quasi-Cauchy, requires less simulations. One should be reminded however that the corresponding computations failed for the non-quadratic problem in the absence of scaling.
- The quasi-Cauchy diagonal-preconditioner update formula was first implemented using Newton-Raphson's unidimensional root-finding algorithm, but there were some failures related to the difficulty of specifying a good problem-independent first estimate of the root. As in Zhu et al. (1999), it was finally implemented using bisection. Quasi-Cauchy performs worse in this case study than the formulae proposed in Gilbert and Lemaréchal (1989). This suggests that the latter was able to accumulate some useful information on the inverse Hessian.
- As in Gilbert and Lemaréchal (1989) scaling the diagonal preconditioner so that it satisfies the quasi-Cauchy relation improves the performance of the minimization, especially when this scaling is done after updating it. However the inverse Hessian approximation is largely damaged by such a scaling.

This fact led us to propose a new approach, where a scaled version of the diagonal preconditioner is used for the minimization but the original (unscaled) one is updated, using the newest pair. This approach allows a good approximation of the inverse Hessian while improving further the minimization performance. It has also some "natural" properties: increasing the storage leads to better minimization performance and inverse Hessian approximation, and the latter improves steadily during the minimization process. These results were obtained both for the quadratic and for the non-quadratic variational data assimilation problems.

However the improvements of the L-BFGS inverse-Hessian quality and the reduction of simulations needed to achieve convergence may well be specific to the problems studied. The proposed approach was also assessed for a large number of unconstrained problems. An improvement was obtained for the MODULOPT and MINPACK-2 problems, but the tests with the CUTE library evidenced a lack of robustness of the method (Veersé and Auroux, 2000).

Our original concern was the quality of the L-BFGS inverse Hessian approximation. Not surprisingly the diagonal-preconditioners that provide good-quality inverse Hessian approximations are also those that lead to good minimization performances, when the

proposed scaling approach is used. The methodology employed in the present study may thus be worthwhile when designing future diagonal-preconditioner update formulae.

Acknowledgments

The first author would like to dedicate this paper to Prof. P. Fabrie of Université de Bordeaux who taught him a few years ago the basics of optimization theory.

References

- C. G. Broyden, "A new double-rank minimization algorithm," *Notices American Math. Soc.* vol. 16, p. 670, 1969.
- P. Courtier, E. Andersson, W. Heckley, G. Kelly, J. Pailleux, F. Rabier, J.-N. Thépaut, P. Uden, D. Vasiljevic, C. Cardinali, J. Eyre, M. Hamrud, J. Haseler, A. Hollingsworth, A. McNally, and A. Stoffelen, "Variational assimilation at ECMWF," ECMWF Research Dept. Tech. Memo. vol. 194, p. 84, 1993.
- P. Courtier, J.-N. Thépaut, and A. Hollingsworth, "A strategy for operational implementation of 4D-Var, using an incremental approach," *Quart. J. Roy. Meteor. Soc.* vol. 120, pp. 1367–1387, 1994.
- J. E. Dennis and J. J. Moré, "Quasi-Newton methods, motivation and theory," *SIAM Rev.* vol. 19, pp. 46–89, 1977.
- J.-Ch. Gilbert and C. Lemaréchal, "Some numerical experiments with variable storage quasi-Newton algorithms," *Math. Prog.* vol. 45, pp. 407–435, 1989.
- F.-X. Le Dimet, H. E. Ngodock, and B. Luong, "Sensitivity analysis in variational data assimilation," *J. Meteorol. Soc. Japan* vol. 75-1B, pp. 245–255, 1997.
- R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicit Restarted Arnoldi Methods*, 1997, Available from <http://www.caam.rice.edu/software/ARPACK/>
- D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Math. Prog.* vol. 45, pp. 503–528, 1989.
- J. Nocedal, "Updating quasi-Newton matrices with limited-storage," *Math. Comput.* vol. 35, pp. 773–782, 1980.
- F. Veersé, "Variable-storage quasi-Newton operators as inverse forecast/analysis error covariance matrices in variational data assimilation," INRIA Research Report 3685, p. 27, 1999.
- F. Veersé, "Variable-storage quasi-Newton operators for modeling error covariances," in *Proceedings of the Third WMO International Symposium on Assimilation of Observations in Meteorology and Oceanography, 7–11 June 1999*, Quebec City, Canada, World Meteorological Organization, Geneva, to appear.
- F. Veersé and D. Auroux, "Some numerical experiments on scaling and updating L-BFGS diagonal preconditioners," INRIA Research Report 3858, p. 25, 2000.
- Z. Wang, I. M. Navon, F.-X. Le Dimet, and X. Zou, "The second order adjoint analysis: Theory and applications," *Meteorol. Atmos. Phys.* vol. 50, pp. 3–20, 1992.
- P. Wolfe, "Convergence conditions for ascent methods," *SIAM Rev.* vol. 11, no. 2, pp. 226–235, 1969.
- M. Zhu, J.-L. Nazareth, and H. Wolkowicz, "The quasi-Cauchy relation and diagonal updating," *SIAM J. Optim.* vol. 9, pp. 1192–1204, 1999.

Annexe 2

Data assimilation methods for an oceanographic problem.

D. Auroux and J. Blum

À paraître dans la série *Mathematics in Industry*, Springer-Verlag.

Data assimilation methods for an oceanographic problem

Didier Auroux¹ and Jacques Blum²

¹ Laboratoire J.A. Dieudonné, Université de Nice Sophia-Antipolis, Parc Valrose, F-06108 Nice Cedex 2, France. auroux@math.unice.fr

² Laboratoire J.A. Dieudonné, Université de Nice Sophia-Antipolis, Parc Valrose, F-06108 Nice Cedex 2, France. jblum@math.unice.fr

1 Introduction

The dynamics of the oceans play a major role in the knowledge of our environment and especially in the Earth's climate. Over the past twenty years, the new satellite techniques for observing the oceans, and especially the use of altimeter measurements, have greatly improved our knowledge of the oceans by allowing synoptic monitoring of the surface. The measurements of the sea-surface height have clearly demonstrated the feasibility and the usefulness of satellite altimetry. It was with the availability of Topex/Poseidon data since 1992, that the oceanographic community began intensive exploitation of this new observational source. It has already given incomparable information to study the general circulation of the ocean, to estimate the energy levels of the upper ocean, and to examine the local dynamics of different regions of particular interest, such as the Gulf Stream area, the Kuroshio extension, the Antarctic circumpolar current and the tropical oceans.

At the interface between the two major components of oceanographic science, i.e. observations and models, lies the domain of so-called data assimilation (DA). DA covers all the mathematical and numerical techniques which allow us to blend as optimally as possible all the sources of information coming from theory, models and other types of data. Clearly these techniques may not only apply in oceanography but also to other environmental disciplines. DA allows us to recreate the time-space structure of a system from a set of information which has, in general, a large disparity in nature, in space-time distribution and in accuracy. There are two main categories of DA methods: variational methods based on the optimal control theory [Lio68] and statistical methods based on the theory of optimal statistical estimation. The prototype of the first class which is actually of interest here is the optimal control method which was first introduced in meteorology (see [Lew85], [LeD86], [Tal87]) and more recently for the ocean (see [Tha88], [She90], [Moo91], [Sch93], [Nec94], [Luo98]). The prototype of statistical methods is the Kalman filter whose introduction

in oceanography dates back roughly a decade (see, for example, [Ghi89] and [Ghi91]). The Kalman filter was extended to nonlinear cases ([Jaz70], [Gel74]) but it has been mostly applied in oceanography to quasi-linear situations of the tropical oceans ([Gou92], [Fuk95], [Fuk93], [Can96], [Ver99]). We also refer to the recent book of Bennett [Ben02] on inverse methods, both for oceanography and meteorology.

All DA techniques encounter major difficulties in practice for computing reasons: memory size and computing costs. The full Kalman filter would, in principle, require the manipulation of $(N \times N)$ matrices where N is the state vector dimension which is typically 10^7 or 10^8 in an oceanic problem. The optimal control adjoint method often requires several hundred iterations of the minimization process to converge, thus implying an equivalent number of model runs.

In this paper, we first focus our interest on the use of the variational adjoint method in a relatively simple ocean model in order to try to reconstruct the four-dimensional ocean system from altimetric surface observations of the ocean. The variational method uses the strong constraint hypothesis, i.e. the ocean circulation model is assumed to be exact. The assimilation process is carried out by an identification of the initial state of the dynamical system which minimizes a cost function. This cost function is the mean-square difference between the observations and the corresponding model variables. The functional will be minimized using a numerical unconstrained optimization method such as the limited memory BFGS algorithm (see [Gil89]). The gradient vector is obtained analytically from the adjoint state, which can be interpreted as the Lagrange multiplier of the model equations. We then use a dual method, which consists in considering the model as a weak constraint. The use of an observation vector as a Lagrange multiplier for this constraint allows us to consider the minimization problem in a dual way. The dual cost function, measuring the difference between the data and the model state corresponding to a vector of the observation space, is minimized in the observation space, still using the BFGS algorithm.

In section 2, we introduce the physical model used for the theoretical and numerical results. The primal and dual methods applied to our ocean model are introduced in sections 3 and 4 respectively. Some numerical results are given in section 5. A few conclusions will be given in section 6.

2 Physical model

2.1 Quasi-geostrophy

The system which governs the behaviour of the ocean is called the primitive equation system, constituted by the conservation laws of mass, momentum (Navier-Stokes equations), temperature and salinity. Most large-scale geophysical flows are based on the geostrophic equilibrium between the rotational

effect due to the Coriolis force and the horizontal pressure gradient. We will use here a simplified quasi-geostrophic ocean model. This model arises from the primitive equations, assuming first that the rotational effect (Coriolis force) is much stronger than the inertial effect. This can be quantified by the fact that the ratio between the characteristic time of the rotation of the Earth and the inertial time is small. This ratio is called the Rossby number. The quasi-geostrophic model also assumes that the size of the ocean is small compared to the size of the Earth, and that this ratio is close to the Rossby number. Quasi-geostrophy finally assumes that the depth of the basin is small compared to its width (the ocean is supposed to be a thin layer of the Earth). In the case of the Atlantic Ocean, all these assumptions are not valid, but it has been shown that this approximate model reproduces quite well the ocean circulations at intermediate latitudes, such as the Gulf Stream. The thermodynamic effects are neglected, and we also assume that the forcing is due to the wind at the surface of the ocean and that the dissipation is essentially due to bottom and lateral friction.

2.2 Equations of the model

The ocean is supposed to be stratified in n layers, each of them having a constant fluid density [Hol78]. The quasi-geostrophic model is obtained by making a first order expansion of the Navier-Stokes equation with respect to the Rossby number [Ped79]. The model system is then composed of n coupled equations resulting from the conservation law of the potential vorticity. The equations can be written as :

$$\frac{D_1(\theta_1(\Psi) + f)}{Dt} - \beta \Delta^2 \Psi_1 = F_1 \quad \text{in } \Omega \times]0, T[, \quad (1)$$

at the surface layer ($k = 1$),

$$\frac{D_k(\theta_k(\Psi) + f)}{Dt} - \beta \Delta^2 \Psi_k = 0 \quad \text{in } \Omega \times]0, T[, \quad (2)$$

at intermediate layers ($k = 2, \dots, n - 1$), and

$$\frac{D_n(\theta_n(\Psi) + f)}{Dt} + \alpha \Delta \Psi_n - \beta \Delta^2 \Psi_n = 0 \quad \text{in } \Omega \times]0, T[, \quad (3)$$

at the bottom layer ($k = n$), where

- $\Omega \subset \mathbb{R}^2$ is the circulation basin and $]0, T[$ is the time interval,
- n is the number of layers,
- Ψ_k is the stream function at layer k , Ψ is the vector $(\Psi_1, \dots, \Psi_n)^T$,

- θ_k is the sum of the dynamical and thermal vorticity at layer k :

$$\theta_k(\Psi) = \Delta\Psi_k - (W\Psi)_k,$$

$$\text{with } -(W\Psi)_k = \frac{f_0^2 \rho}{H_k g} \left(\frac{\Psi_{k+1} - \Psi_k}{\rho_{k+1} - \rho_k} - \frac{\Psi_k - \Psi_{k-1}}{\rho_k - \rho_{k-1}} \right).$$

- f is the Coriolis force (f_0 is the Coriolis force at the reference latitude of the ocean).
In the β -plane approximation, the Coriolis force varies linearly with respect to the latitude.
- g represents the gravity, ρ_k the fluid density at layer k (and ρ the average fluid density), and H_k the depth of the layer k ,
- $\frac{D_k}{Dt}$ is the Lagrangian particular derivative : $\frac{D_k}{Dt} = \frac{\partial}{\partial t} + J(\Psi_k, \cdot)$,
where J is the Jacobian operator $J(f, g) = \frac{\partial f}{\partial x} \frac{\partial g}{\partial y} - \frac{\partial f}{\partial y} \frac{\partial g}{\partial x}$,
- $\Delta\Psi_n$ represents the bottom friction dissipation, $\Delta^2\Psi_k$ represents the lateral friction dissipation,
- and F_1 is the forcing term, the wind stress applied to the ocean surface.

2.3 Boundary conditions

The tridiagonal matrix W (used to couple the stream functions at different layers) can be diagonalized :

$$W = P \cdot \text{diag}(\lambda_1, \dots, \lambda_n) \cdot P^{-1}, \quad (4)$$

where $0 = \lambda_1 < \lambda_2 \leq \dots \leq \lambda_n$ are the eigenvalues, and P is the transformation matrix. We can then define the mode vector of the stream functions $\Phi = (\Phi_1, \dots, \Phi_n)^T$:

$$\Phi = P^{-1}\Psi.$$

The first mode Φ_1 corresponds to the eigenvalue 0 and is called the barotropic mode. The next modes are the baroclinic modes. The boundary conditions result from the mass conservation law (Holland 1978), and can then be written as :

$$\begin{aligned} \Phi_1 &= 0 && \text{in } \partial\Omega \times]0, T[, \\ \int_{\Omega} \Phi_k(t) d\sigma &= 0 && \forall t \in [0, T], \quad \forall k \geq 2, \end{aligned} \quad (5)$$

and

$$\Delta\Psi_k(t) = 0 \quad \text{in } \partial\Omega \times]0, T[, \quad \forall k. \quad (6)$$

The initial conditions $\Psi_k(0)$ complete the equations of the direct model.

3 Primal variational method

We suppose that the data we want to assimilate come from satellite measurements of the sea-surface height, which is directly related to the upper layer stream function Ψ_1 by $h = \frac{f_0}{g}\Psi_1$. Thus, we assume that we have an observational stream function Ψ_1^{obs} . These observations are only available at times t_i , $i = 1 \dots N$, over the data assimilation period $[0, T]$, and are also discrete in space. We consider then that the vector $\Psi_1^{obs}(t_i)$ represents the observations of the ocean surface available at time t_i . The control vector u (which has to be determined) is the initial state of the stream functions at all layers $(\Psi_k(0))_{k=1\dots n}$.

3.1 Cost function

We can define a cost function

$$\begin{aligned} \mathcal{J}(u) = & \frac{1}{2} \sum_{i=1}^N \langle R_i^{-1} (H_i \Psi_1(t_i) - \Psi_1^{obs}(t_i)), H_i \Psi_1(t_i) - \Psi_1^{obs}(t_i) \rangle \\ & + \frac{1}{2} \langle P_0^{-1} u, u \rangle, \end{aligned} \quad (7)$$

where P_0 and R_i are covariance matrices, H_i are (linear) observation operators connecting observations Ψ_1^{obs} and model solutions Ψ_1 , and $\langle \cdot, \cdot \rangle$ is the canonical real scalar product.

The first part of the cost function quantifies the difference between the observations and the state function, and the second part is a regularisation term. The inverse problem which consists in the minimization of \mathcal{J} is then well-posed.

3.2 Adjoint model

In order to minimize the cost function, we need its gradient $\nabla \mathcal{J}$. Because of the large dimension of the model state vector (more than 10^6), it is not possible to compute directly the gradient by using finite difference methods. The gradient vector of the functional is then obtained by solving backwards in time the adjoint model ([LeD86]). The quasi-geostrophic adjoint equations are :

$$\begin{aligned} \frac{\partial \theta_1^T(\Lambda)}{\partial t} - \Delta J(\Psi_1, \Lambda_1) - (W^T J(\Psi, \Lambda))_1 - J(\Lambda_1, \theta_1(\Psi) + f) \\ - \beta \Delta^2 \Lambda_1 = E_1 \end{aligned} \quad (8)$$

at the surface layer,

$$\frac{\partial \theta_k^T(\Lambda)}{\partial t} - \Delta J(\Psi_k, \Lambda_k) - (W^T J(\Psi, \Lambda))_k - J(\Lambda_k, \theta_k(\Psi) + f) - \beta \Delta^2 \Lambda_k = 0 \quad (9)$$

at the intermediate layers, and

$$\frac{\partial \theta_n^T(\Lambda)}{\partial t} - \Delta J(\Psi_n, \Lambda_n) - (W^T J(\Psi, \Lambda))_n - J(\Lambda_n, \theta_n(\Psi) + f) + \alpha \Delta \Lambda_n - \beta \Delta^2 \Lambda_k = 0 \quad (10)$$

at the bottom layer, in $\Omega \times]0, T[$, where

- $\Lambda_1, \dots, \Lambda_n$ is the adjoint vector,
- $\theta_k^T(\Lambda) = -\Delta \Lambda_k + (W^T \Lambda)_k$ is the *vorticity* corresponding to the adjoint state,
- and E_1 is the derivative of \mathcal{J} with respect to Ψ_k :

$$E_1(t) = \sum_{i=1}^N R_i^{-1} (H_i \Psi_1(t) - \Psi_1^{obs}(t)) \delta(t - t_i).$$

If we denote by $\chi = (\chi_1, \dots, \chi_n)^T$ the modal adjoint vector :

$$\chi = P^T \Lambda,$$

the space boundary conditions satisfied by the adjoint state Λ are :

$$\begin{aligned} \chi_1 &= 0 && \text{in } \partial\Omega \times]0, T[, \\ \int_{\Omega} \chi_k(t) d\sigma &= 0 && \forall t \in [0, T], \quad \forall k \geq 2, \end{aligned} \quad (11)$$

and

$$\Delta \Lambda_k(t) = 0 \quad \text{in } \partial\Omega \times]0, T[, \quad \forall k. \quad (12)$$

The gradient of the first part of \mathcal{J} is obtained by solving equations (8-12) with a final condition of nullity of the adjoint state. The gradient of the second part of \mathcal{J} is obtained directly by deriving it with respect to u , and we obtain :

$$\nabla \mathcal{J} = H(-\Delta + W)H^{-1} \begin{pmatrix} \Lambda_1(0) \\ \vdots \\ \Lambda_n(0) \end{pmatrix} + P_0^{-1} u \quad (13)$$

where H is the diagonal matrix with the layers' depths H_k on the diagonal.

3.3 Minimization process

The numerical minimization of the cost function \mathcal{J} can be realized using a quasi-Newton method. The Newton class of minimization algorithms is based on an iterative process, using at iteration k the descent direction $d_k = -H_k^{-1} \cdot \nabla \mathcal{J}(x_k)$, where $H_k = \nabla^2 \mathcal{J}(x_k)$ is the Hessian matrix of the cost function. The direct computation of the Hessian matrix is impossible (its dimension being too large), but it is possible to evaluate it, using the second order adjoint equations. However, the inversion of H is nearly impossible. The quasi-Newton algorithms consist in replacing the inverse of the Hessian matrix by a succession of matrices $(W_k)_{k \in \mathbb{N}}$ which are symmetric positive definite approximations to H^{-1} .

The BFGS algorithm ([Bro69]) uses the following update formula :

$$W_{k+1} = U(W_k, s_k, \eta_k) := \left(I - \frac{s_k \otimes \eta_k}{\langle \eta_k, s_k \rangle} \right) W_k \left(I - \frac{\eta_k \otimes s_k}{\langle \eta_k, s_k \rangle} \right) + \frac{s_k \otimes s_k}{\langle \eta_k, s_k \rangle}$$

with $s_k = x_{k+1} - x_k$, $\eta_k = \nabla \mathcal{J}(x_{k+1}) - \nabla \mathcal{J}(x_k)$ and $a \otimes b : c \mapsto \langle b, c \rangle a$. The disadvantage of this formula is the need to store all pairs (s_k, η_k) .

The L-BFGS algorithm ([Liu89]) is a limited memory version of the previous algorithm. Only the last M pairs are stored, M being often equal to 5. The update formula is then :

$$W_k = U(W_{k-1}, s_{k-1}, \eta_{k-1}), \quad 1 \leq k \leq M,$$

and

$$\begin{cases} W_k^0 = D_k, \\ W_k^{i+1} = U(W_k^i, s_{k-M+i}, \eta_{k-M+i}), & i = 0 \dots M-1, \\ W_k = W_k^M, \end{cases}$$

for $k \geq M+1$, where D_k is a diagonal matrix. The update formula for D_k is :

$$D_{k+1}^{(i)} = \left(\frac{1}{D_k^{(i)}} + \frac{\eta_k^{(i)2}}{\langle \eta_k, s_k \rangle} - \frac{s_k^{(i)2}}{(D_k^{(i)})^2 \langle D_k^{-1} s_k, s_k \rangle} \right)^{-1}.$$

4 Dual method

4.1 General description

The primal method has many disadvantages. First, the minimization process is often stopped before convergence to the minimum, because of the size of the state vector. Moreover, it is also impossible to take into account a model error : in the previous section, we have supposed that the model and the equations were perfect. This is obviously not the case (for example, not all parameters are well known). The only solution to incorporate the model error

into the minimization process is to add corrective terms to the model, consider them as part of the control vector, and add a third term to the cost function. This is not computationally realistic because the size of the control vector would be multiplied by the number of time steps. Therefore, it is not possible to take into account in a straightforward way the model error in the primal variational approach.

A new approach to data assimilation problems has been recently introduced ([Amo95], [Ben92], [Cou97]). Rather than minimizing a cost function on the state space, the dual method consists in working in the observation space (which is smaller than the state space).

4.2 Dual algorithm

Instead of solving first the direct equations and then the adjoint equations in the primal variational approach, the dual method consists in solving first the adjoint equations in order to use the information contained in the observation vector, and then the direct equations in order to reconstruct a trajectory. The dual algorithm for the quasi-geostrophic model can be constructed as follows :

- Let m be an observation vector that can be directly related to Ψ_1 (assume that m is a vector containing an observation of a part of the ocean surface at different times t_i),
- Solve the adjoint equations (with a final condition equal to zero) :

$$\begin{aligned}
\frac{\partial \theta_1^T(\Lambda)}{\partial t} - \Delta J(\Psi_1, \Lambda_1) - (W^T J(\Psi, \Lambda))_1 - J(\Lambda_1, \theta_1(\Psi) + f) \\
- \beta \Delta^2 \Lambda_1 = \tilde{E}_1(m), \\
\frac{\partial \theta_k^T(\Lambda)}{\partial t} - \Delta J(\Psi_k, \Lambda_k) - (W^T J(\Psi, \Lambda))_k - J(\Lambda_k, \theta_k(\Psi) + f) \\
- \beta \Delta^2 \Lambda_k = 0, \quad 1 < k < n, \\
\frac{\partial \theta_n^T(\Lambda)}{\partial t} - \Delta J(\Psi_n, \Lambda_n) - (W^T J(\Psi, \Lambda))_n - J(\Lambda_n, \theta_n(\Psi) + f) \\
+ \alpha \Delta \Lambda_n - \beta \Delta^2 \Lambda_k = 0,
\end{aligned} \tag{14}$$

where

$$\tilde{E}_1(m)(t) = \sum_{i=1}^N H_i^T R_i^{-1} (m(t) - \Psi_1^{obs}(t)) \delta(t - t_i).$$

- Solve the direct equations

$$\begin{aligned}
\frac{D_1(\theta_1(\Psi) + f)}{Dt} - \beta\Delta^2\Psi_1 &= F_1 + (QA)_1, \\
\frac{D_k(\theta_k(\Psi) + f)}{Dt} - \beta\Delta^2\Psi_k &= (QA)_k, \quad 1 < k < n \\
\frac{D_n(\theta_n(\Psi) + f)}{Dt} + \alpha\Delta\Psi_n - \beta\Delta^2\Psi_n &= (QA)_n,
\end{aligned} \tag{15}$$

with the initial conditions

$$\Psi_k(0) = \Psi_k^e(0) + (P_0A(0))_k,$$

where Q and P_0 are statistical preconditioning matrices, and $\Psi_k^e(0)$ is an *a priori* estimation of $\Psi_k(0)$.

- Define the operator $\mathcal{D} : (\mathcal{D}m)(t) = \sum_{i=1}^N H_i\Psi_1(t_i) \delta(t - t_i)$.

We can then define the dual cost function as follows :

$$\mathcal{J}_{\mathcal{D}}(m) = \frac{1}{2}\langle \mathcal{D}m, m \rangle - \langle \Psi_1^{obs}, m \rangle. \tag{16}$$

$\mathcal{J}_{\mathcal{D}}$ measures the difference between $\mathcal{D}m$ and Ψ_1^{obs} , i.e. between the trace (in the observation space) of a solution of the direct model and the observation vector.

As \mathcal{D} is a linear symmetric positive definite operator, the gradient is obviously given by

$$\nabla \mathcal{J}_{\mathcal{D}}(m) = \mathcal{D}m - \Psi_1^{obs}. \tag{17}$$

It is therefore easy to perform the minimization of $\mathcal{J}_{\mathcal{D}}$, given its gradient, simply by using a quasi-Newton method such as a BFGS algorithm. Once the minimum has been found, it is easy to reconstruct the corresponding trajectory in the state space by solving (14-15).

We can observe that the minimization of the dual cost function takes place over a smaller space than the minimization of the primary one. Moreover, this method also takes into account the model error, which was numerically impossible in the classical approach.

5 Numerical results

5.1 Model parameters

The numerical experiments are performed for a square three-layered ocean. The basin has horizontal dimensions of 4000 km \times 4000 km and its depth is 5 km. The layers' depths are 300 meters for the surface layer, 700 meters for the intermediate layer, and 4000 meters for the bottom layer. The ocean is

discretized by a Cartesian mesh of $200 \times 200 \times 3$ grid zones. The time step is 1.5 hour. The initial conditions are chosen equal to zero for a six-year ocean spin-up phase, the final state of which being then the initial state for the data assimilation period. Then the assimilation period starts (time $t = 0$) with this initial condition ($\Psi_k(0)$), and lasts 5 days (time $t = T$), i.e. 80 time steps. The numerical method used to minimize the cost functions is a limited memory BFGS quasi-Newton method. The M1QN3 code by Gilbert and Lemaréchal ([Gil89]) is used for our experiments.

The experimental approach consists in performing *twin experiments* with simulated data. First, a reference experiment is run and the corresponding data are extracted. This reference trajectory will be further called the exact solution. Experimental surface data are supposed to be obtained on every fifth gridpoint of the model, with a time sampling of 7.5 hours (every 5 time steps). Simulated surface data are then noised with a blank Gaussian distribution, and provided as observations for the cost function. The first guess of the assimilation experiments is chosen as the reference state of the ocean one year before the assimilation period. The results of the identification process are then compared to the reference experiment.

5.2 Exact solution, noised observations

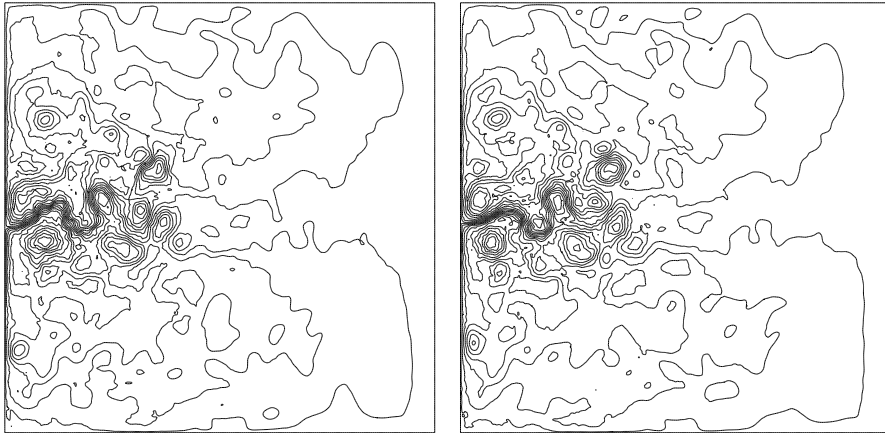


Fig. 1. Exact solution at the beginning (a), resp. the end (b), of the assimilation period.

Fig. 1 represents the stream function Ψ_1 at the surface layer, at the beginning and at the end of the assimilation period. These fields will be useful to measure the identification of the initial state, and also the reconstruction of

the stream function at the final time. One can observe the turbulent structure of the ocean, with a main current simulating a Gulf Stream type configuration.



Fig. 2. Noised extracted data at the surface layer (a) and corresponding state at the end of the assimilation period (b)

The first part of fig. 2 represents the noised data extracted from the reference run, still at the surface layer. The second part of this figure is the corresponding state after a model run using the noised data as initial condition. This experiment clearly shows the importance of data assimilation. The model will indeed not smooth the trajectory, and it is not possible to obtain good predictions by simply integrating the model with observation data as initial conditions.

5.3 Primal method

The initial estimated vector to start the minimization process is chosen to be the reference state of the ocean one year before the assimilation period. The minimization process is stopped after 40 iterations, each iteration consisting of one integration of the forward direct model (in order to compute \mathcal{J}) and one integration of the backward adjoint model (in order to compute $\nabla\mathcal{J}$). The result of the minimization is shown on fig. 3-a. The direct model is then integrated over the assimilation period, using the computed minimizer as initial condition, and the corresponding state of the ocean at the end of the assimilation period is shown on fig. 3-b.

We can notice that the stream function of the solution at time $t = 0$ at the surface layer is comparable to the exact solution at the same time, but to a

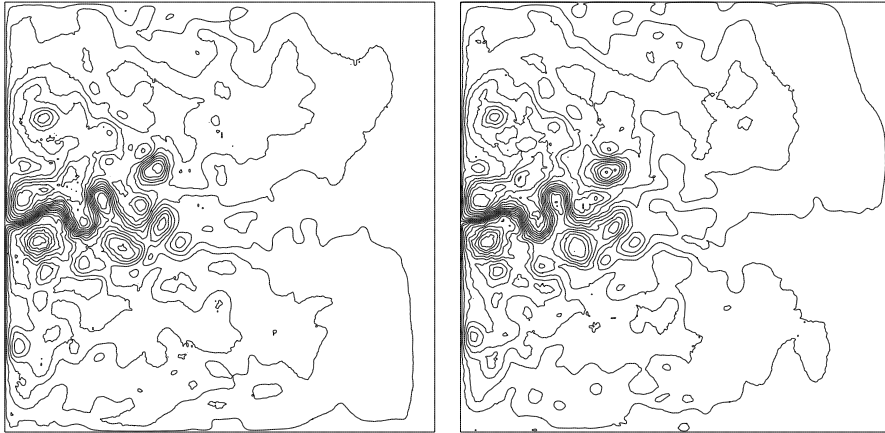


Fig. 3. Result of the minimization of the primal cost function. Solution at the beginning (a) and the end (b) of the assimilation period

lesser extent at time $t = T$. This can be explained by the fact that the primal algorithm gives more importance to the state at $t = 0$ than to any other time, as it is the control vector.

5.4 Dual method

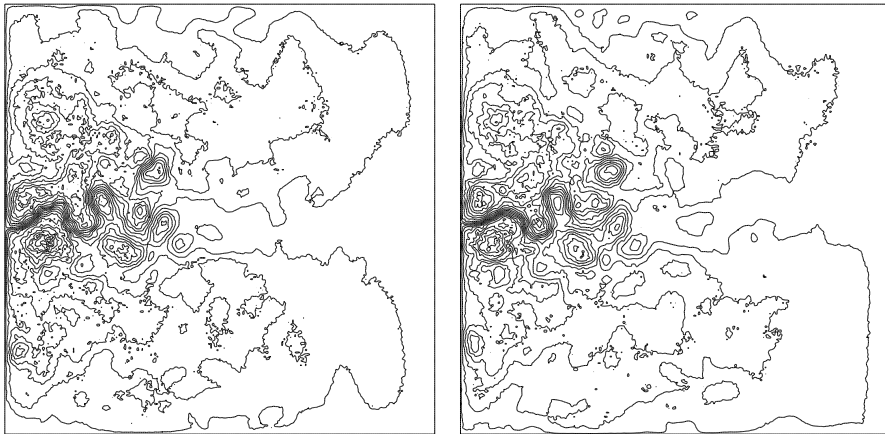


Fig. 4. Result of the minimization of the dual cost function. Solution at the beginning (a) and the end (b) of the assimilation period

The initial estimated vector is the same. The minimization process is still stopped after 40 iterations, each iteration consisting now of one integration of the backward adjoint equations and one integration of the direct equations (in order to compute $\mathcal{J}_{\mathcal{D}}$ and $\nabla \mathcal{J}_{\mathcal{D}}$). The result of the minimization process is shown on fig. 4-a at time $t = 0$ and fig. 4-b at time $t = T$.

The stream function appears to be less smooth than in the primal case. This is due to the fact that the observations are noised and the dual algorithm works over the observation space. The corresponding state at the end of the assimilation period is closer to the exact solution than in the primal case. The dual algorithm looks indeed for a global acceptable solution : the control vector is a set of *observations* all over the assimilation period.

5.5 Comparison between the two methods

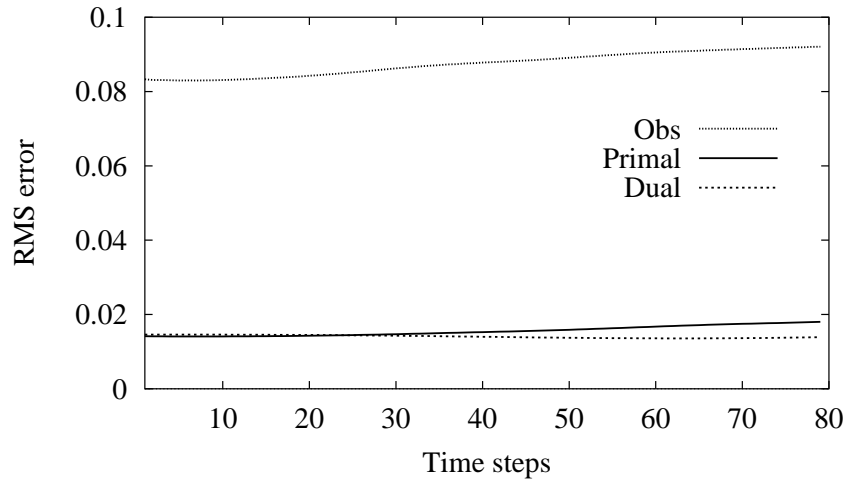


Fig. 5. RMS errors of the different methods versus number of time steps over the assimilation period, using as initial conditions: the first noised observation (thin line), the primal algorithm (bold line) and the dual algorithm (dot line).

Fig. 5 represents the root mean square (RMS) error over the entire assimilation period between the exact solution and an identified solution, using either the observations, the primal solution, or the dual one. The RMS error at time t for one of these solutions is :

$$rms(t) = \frac{\int_{\Omega} [\Psi_1^{\text{sol}}(t) - \Psi_1^{\text{exact}}(t)]^2 d\sigma}{\int_{\Omega} [\Psi_1^{\text{exact}}(t)]^2 d\sigma}.$$

The error reaches nearly 10% in the case of the trajectory resulting from the observations, and increases in time. This is due to the inherent non-linearities of the model.

In the case of the two data assimilation methods, the RMS error is clearly smaller (by a factor of more than 5). This proves once again the usefulness of data assimilation, which makes it possible to reconstruct a trajectory with less than 2% RMS error using noised observations with a 10% RMS error. The RMS error of the primal method tends to increase in time. This is in agreement with the remark concerning the resemblance between the stream function of the primal solution and that of the exact solution at initial time, and the loss of this similarity at final time. The RMS error of the dual solution is a little larger, but it tends to remain constant (and even decrease a little bit sometimes) in time. Moreover, as the dimension of the control vector u in the primal variational approach is 121203 ($201 \times 201 \times 3$), versus 28577 in the dual one ($41 \times 41 \times 17$), the minimization of the dual cost function is faster than for the primal approach.

6 Conclusion

As a matter of fact, the oceanic circulation model is governed by complex equations and behave as certain typical characteristics of the turbulent flow. Besides, in practice, the observation data are of various nature and should be combined together in the same functional to be minimized. In the present work, synthetic data are sampled using the whole surface layer. Generally, in the framework of the realistic oceanic data assimilation, the data are available only along ground tracks for time intervals corresponding to the satellite repeat period. Therefore, the optimal initial state would not be as well estimated because of the relatively small number of observations and their heterogeneous spatial distribution. Also, we notice that the performances of these methods have been assessed with a quasi-geostrophic model. It is necessary to apply them to a more complicated model such as the primitive equation model.

The dual data assimilation method is promising taking into account the computing time which is smaller than the primal optimal control method because of the smaller dimension of the observation space. Moreover, it enables to introduce an error in the model and not to consider the equations of the fluid as a strong constraint. The non linear character of the equations remains a problem for the proof of convergence of the dual method.

It remains a promising step towards operational oceanography.

References

- [Amo95] Amodei, L.: Solution approchée pour un problème d'assimilation de données météorologiques avec prise en compte de l'erreur modèle. C.R. Acad. Sci. Paris, **321 (II)**, 1087–1094 (1995)

- [Ben92] Bennett, A.F.: Inverse methods in physical oceanography. Cambridge University Press, Cambridge (1992)
- [Ben02] Bennett, A.F.: Inverse Modeling of the Ocean and Atmosphere. Cambridge University Press, Cambridge (2002)
- [Bro69] Broyden, C.G.: A new double-rank minimization algorithm. *Notices American Math. Soc.*, **16**, 670 (1969)
- [Can96] Cane, M.A., Kaplan, A., Miller, R.N., Tang, B., Hackert, E.C., Busalacchi, A.J.: Mapping tropical Pacific sea level: data assimilation via a reduced state Kalman filter. *J. Geophys. Res.*, **101**(C10), 22599–22617 (1996)
- [Cou97] Courtier, P.: Dual formulation of four-dimensional variational assimilation. *Q. J. R. Meteorol. Soc.*, **123**, 2449–2461 (1997)
- [Fuk93] Fukumori, I., Benveniste, J., Wunsch, C., Haidvogel, D.B.: Assimilation of sea surface topography into an ocean circulation model using a steady state smoother. *J. Phys. Oceanogr.*, **23**, 1831–1855 (1993)
- [Fuk95] Fukumori, I.: Assimilation of Topex sea level measurements with a reduced-gravity, shallow water model of the tropical Pacific ocean. *J. Geophys. Res.*, **100**(C12), 25027–25039 (1995)
- [Gel74] Gelb, A.: Applied Optimal Estimation. MA: MIT Press, Cambridge (1974)
- [Ghi89] Ghil, M.: Meteorological data assimilation for oceanographers. Part I: description and theoretical framework. *Dyn. Atmos. Oceans*, **13**, 171–218 (1989)
- [Ghi91] Ghil, M., Manalotte-Rizzoli, P.: Data assimilation in meteorology and oceanography. *Adv. Geophys.*, **23**, 141–265 (1991)
- [Gil89] Gilbert, J.-Ch., Lemaréchal, C.: Some numerical experiments with variable storage quasi-Newton algorithms. *Math. Prog.*, **45**, 407–435 (1989)
- [Gou92] Gourdeau, L., Arnault, S., Ménard, Y., Merle, J.: Geosat sea-level assimilation in a tropical Atlantic model using Kalman filter. *Ocean. Acta*, **15**, 567–574 (1992)
- [Hol78] Holland, W.R.: The role of mesoscale eddies in the general circulation of the ocean. *J. Phys. Oceanogr.*, **8-3**, 363–392 (1978)
- [Jaz70] Jazwinski, A.H.: Stochastic Processes and Filtering Theory. Academic, New York (1970)
- [LeD86] Le Dimet, F.-X., Talagrand, O.: Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus*, **38A**, 97–110 (1986)
- [Lew85] Lewis, J.M., Derber, J.C.: The use of adjoint equations to solve a variational adjustment problem with convective constraints. *Tellus*, **37A**, 309–322 (1985)
- [Lio68] Lions, J.L.: Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles. Dunod (1968)
- [Liu89] Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. *Math. Prog.*, **45**, 503–528 (1989)
- [Luo98] Luong, B., Blum, J., Verron, J.: A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, **14**, 979–997 (1998)
- [Moo91] Moore, A.M.: Data assimilation in a quasigeostrophic open-ocean model of the Gulf-Stream region using the adjoint model. *J. Phys. Oceanogr.*, **21**, 398–427 (1991)
- [Nec94] Nechaev, V., Yaremchuk, M.I.: Application of the adjoint technique to processing of a standard section data set: world ocean circulation experiment section S4 along 67S in the Pacific ocean. *J. Geophys. Res.*, **100**(C1), 865–879 (1994)

- [Ped79] Pedlosky, J.: Geophysical fluid dynamics. Springer-Verlag, New-York (1979)
- [Sch93] Schröter, J., Seiler, U., Wenzel, M.: Variational assimilation of Geosat data into an eddy-resolving model of the Gulf Stream area. *J. Phys. Oceanogr.*, **23**, 925–953 (1993)
- [She90] Sheinbaum, J., Anderson, D.L.T.: Variational assimilation of XBT data. Part I. *J. Phys. Oceanogr.*, **20**, 672–688 (1990)
- [Tal87] Talagrand, O., Courtier, P.: Variational assimilation of meteorological observations with the adjoint vorticity equation. Part I: Theory. *Q. J. R. Meteorol. Soc.*, **113**, 1311–1328 (1987)
- [Tha88] Thacker, W.C., Long, R.B.: Fitting dynamics to data. *J. Geophys. Res.*, **93**, 1227–1240 (1988)
- [Ver99] Verron, J., Gourdeau, L., Pham, D.T., Murtugudde, R., Busalacchi, A.J.: An extended Kalman filter to assimilate satellite altimeter data into a non-linear numerical model of the tropical Pacific Ocean: method and validation. *J. Geophys. Res.*, **104**, 5441–5458 (1999)

Bibliographie

- [1] L. Amodei. Solution approchée pour un problème d'assimilation de données météorologiques avec prise en compte de l'erreur modèle. *C.R. Acad. Sci. Paris*, t. 321, série II : 1087–1094, 1995.
- [2] E. Andersson. The ecmwf implementation of the three-dimensional variational assimilation (3d-var), iii : Experimental results. *Q. J. R. Meteorol. Soc.*, 124 : 1831–1860, 1998.
- [3] R.A. Anthes. Data assimilation and initialization of hurricane prediction models. *J. Atmos. Sci.*, 31 : 702–719, 1974.
- [4] A.F. Bennett. *Inverse methods in physical oceanography*. Cambridge University Press, Cambridge, 1992.
- [5] A.F. Bennett. *Inverse Modeling of the Ocean and Atmosphere*. Cambridge University Press, Cambridge, 2002.
- [6] L. Bourgeois. Contrôle optimal et problèmes inverses en plasticité. Thèse de l'Ecole Polytechnique, 1998.
- [7] C.G. Broyden. A new double-rank minimization algorithm. *Notices American Math. Soc.*, 16 : 670, 1969.
- [8] M.A. Cane, A. Kaplan, R.N. Miller, B. Tang, E.C. Hackert, and A.J. Busalacchi. Mapping tropical pacific sea level : data assimilation via a reduced state kalman filter. *J. Geophys. Res.*, 101 (C10) : 22599–22617, 1996.
- [9] P. Courtier. Dual formulation of four-dimensional variational assimilation. *Q. J. R. Meteorol. Soc.*, 123 : 2449–2461, 1997.
- [10] P. Courtier, E. Andersson, W. Heckley, J. Pailleux, D. Vasiljevic, M. Hamrud, A. Hollingsworth, Rabier F., and M. Fisher. The ecmwf implementation of the three-dimensional variational assimilation (3d-var), i : Formulation. *Q. J. R. Meteorol. Soc.*, 124 : 1783–1807, 1998.
- [11] P. Courtier, J.N. Thépaut, and A. Hollingsworth. A strategy for operational implementation of 4d-var using an incremental approach. *Q. J. R. Meteorol. Soc.*, 120 : 1367–1387, 1994.

- [12] P. De Mey and A.R. Robinson. Assimilation of altimeter eddy fields in a limited-area quasi-geostrophic model. *J. Phys. Oceanogr.*, 17 : 365–384, 1991.
- [13] J.E. Dennis and J.J. Moré. Quasi-newton methods, motivation and theory. *Siam Rev.*, 19 : 46–89, 1977.
- [14] I. Fukumori. Assimilation of topex sea level measurements with a reduced-gravity, shallow water model of the tropical pacific ocean. *J. Geophys. Res.*, 100 (C12) : 25027–25039, 1995.
- [15] I. Fukumori, J. Benveniste, C. Wunsch, and D.B. Haidvogel. Assimilation of sea surface topography into an ocean circulation model using a steady state smoother. *J. Phys. Oceanogr.*, 23 : 1831–1855, 1993.
- [16] A. Gelb. *Applied Optimal Estimation*. MA : MIT Press, Cambridge, 1974.
- [17] M. Ghil. Meteorological data assimilation for oceanographers. part i : description and theoretical framework. *Dyn. Atmos. Oceans*, 13 :171–218, 1989.
- [18] M. Ghil and P. Manalotte-Rizzoli. Data assimilation in meteorology and oceanography. *Adv. Geophys.*, 23 :141–265, 1991.
- [19] J.Ch. Gilbert and C. Lemaréchal. Some numerical experiments with variable storage quasi-newton algorithms. *Math. Prog.*, 45 : 407–435, 1989.
- [20] L. Gourdeau, S. Arnault, Y. Ménard, and J. Merle. Geosat sea-level assimilation in a tropical atlantic model using kalman filter. *Ocean. Acta*, 15 : 567–574, 1992.
- [21] W.R. Holland. The role of mesoscale eddies in the general circulation of the ocean. *J. Phys. Ocean.*, 8-3 : 363–392, 1978.
- [22] A.H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic, New York, 1970.
- [23] M.V. Klibanov and F. Santosa. A computational quasi-reversibility method for cauchy problems for laplace’s equation. *SIAM J. Appl. Math.*, 51(6) : 1653–1675, 1991.
- [24] R. Lattès and J.L. Lions. *Méthode de quasi-réversibilité et applications*. Dunod, Paris, 1967.
- [25] F.X. Le Dimet and O. Talagrand. Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus*, 38A : 97–110, 1986.
- [26] R.B. Lehoucq, D.C. Sorensen, and C. Yang. Arpack user’s guide : solution of large-scale eigenvalue problems with implicit restarted arnoldi methods. <http://www.caam.rice.edu/software/ARPACK/>, 1997.
- [27] J.M. Lewis and J.C. Derber. The use of adjoint equations to solve a variational adjustment problem with convective constraints. *Tellus*, 37A : 309–322, 1985.

- [28] J.L. Lions. *Equations différentielles opérationnelles et problèmes aux limites*. Springer-Verlag, Berlin, 1961.
- [29] J.L. Lions. *Contrôle optimal de systèmes gouvernés par des équations aux dérivées partielles*. Dunod, Paris, 1968.
- [30] D.C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Math. Prog.*, 45 : 503–528, 1989.
- [31] A.C. Lorenc, R.S. Bell, and B. Macpherson. The meteorological office analysis correction data assimilation scheme. *Q. J. R. Meteorol. Soc.*, 117 : 59–89, 1991.
- [32] S. Louvel. Etude d’un algorithme d’assimilation variationnelle de données à contrainte faible. mise en œuvre sur le modèle océanique aux équations primitives micom. Thèse de l’Université Paul Sabatier, 1999.
- [33] S. Louvel. Implementation of a dual variational algorithm for assimilation of synthetic altimeter data in the oceanic primitive equation model micom. *J. Geophys. Res.*, 106 : 9199–9212, 2001.
- [34] B. Luong. Techniques de contrôle optimal pour un modèle quasi-géostrophique de circulation océanique. application à l’assimilation variationnelle des données altimétriques satellitaires. Thèse de l’Université Joseph Fourier Grenoble 1, 1995.
- [35] B. Luong, J. Blum, and J. Verron. A variational method for the resolution of a data assimilation problem in oceanography. *Inverse Problems*, 14 : 979–997, 1998.
- [36] C. Maes, M. Benkiran, and P. De Mey. Sea level comparison between topex/poseidon altimetric data and a global ocean circulation model from an assimilation perspective. *J. Geophys. Res.*, 104 : 15575–15585, 1999.
- [37] A.M. Moore. Data assimilation in a quasigeostrophic open-ocean model of the gulf-stream region using the adjoint model. *J. Phys. Oceanogr.*, 21 : 398–427, 1991.
- [38] V. Nechaev and M.I. Yaremchuk. Application of the adjoint technique to processing of a standard section data set : world ocean circulation experiment section s4 along 67°s in the pacific ocean. *J. Geophys. Res.*, 100 (C1) : 865–879, 1994.
- [39] J. Nocedal. Updating quasi-newton matrices with limited-storage. *Math. Comput.*, 35 : 773–782, 1980.
- [40] J. Pedlosky. *Geophysical fluid dynamics*. Springer-Verlag, New-York, 1979.
- [41] D.T. Pham, J. Verron, and M.C. Roubaud. A singular evolutive extended kalman filter for data assimilation in oceanography. *Inverse Problems*, 14 : 979–997, 1998.
- [42] N.A. Phillips. On the completeness of a multi-variate optimum interpolation for large-scale meteorological analysis. *Monthly Weather Review*, 110 : 1329–1334, 1982.

- [43] E. Polak. *Optimization - Algorithms and Consistent Approximations*. Springer-Verlag, New-York, 1997.
- [44] F. Rabier, A. McNally, E. Andersson, P. Courtier, P. Unden, A. Hollingsworth, and F. Bouttier. The ecmwf implementation of the three-dimensional variational assimilation (3d-var), ii : Structure functions. *Q. J. R. Meteorol. Soc.*, 124 : 1809–1829, 1998.
- [45] J. Schroter, U. Seiler, and M. Wenzel. Variational assimilation of geosat data into an eddy-resolving model of the gulf stream area. *J. Phys. Oceanogr.*, 23 : 925–953, 1993.
- [46] J. Sheinbaum and D.L.T. Anderson. Variational assimilation of xbt data. part i. *J. Phys. Oceanogr.*, 20 : 672–688, 1990.
- [47] O. Talagrand. Variational assimilation - adjoint equations. Data Assimilation for the Earth System, Proceedings, Advanced Study Institute (NATO), Acquafredda di Maratea, Italy, May-June 2002, Kluwer Academic Publishers, 2003.
- [48] O. Talagrand and P. Courtier. Variational assimilation of meteorological observations with the adjoint vorticity equation. part i : Theory. *Q. J. R. Meteorol. Soc.*, 113 : 1311–1328, 1987.
- [49] W.C. Thacker and R.B. Long. Fitting dynamics to data. *J. Geophys. Res.*, 93 : 1227–1240, 1988.
- [50] F. Veersé and D. Auroux. Some numerical experiments on scaling and updating l-bfgs diagonal preconditioners. *INRIA Research Report*, 3858 : 1–25, 2000.
- [51] F. Veersé, D. Auroux, and M. Fisher. Limited-memory bfgs diagonal preconditioners for a data assimilation problem in meteorology. *Optimization and Engineering*, 1 (3) : 323–339, 2000.
- [52] J. Verron, L. Gourdeau, D.T. Pham, R. Murtugudde, and A.J. Busalacchi. An extended kalman filter to assimilate satellite altimeter data into a non-linear numerical model of the tropical pacific ocean : method and validation. *J. Geophys. Res.*, 104 : 5441–5458, 1999.
- [53] P.A. Vidard. Vers une prise en compte des erreurs modèle en assimilation de données 4d-variationnelle. application à un modèle réaliste d’océan. Thèse de l’Université Joseph Fourier Grenoble 1, 2001.
- [54] P. Wolfe. Convergence conditions for ascent methods. *SIAM Rev.*, 11/2 : 226–235, 1969.
- [55] M. Zhu, J.L. Nazareth, and H. Wolkowicz. The quasi-cauchy relation and diagonal updating. *SIAM J. Optim.*, 9 : 1192–1204, 1999.
- [56] X. Zou, I.M. Navon, and F.X. Le Dimet. An optimal nudging data assimilation scheme using parameter estimation. *Q. J. R. Meteorol. Soc.*, 118 : 1163–1186, 1992.

Résumé

L'assimilation de données consiste à combiner de façon optimale les observations d'un système et la connaissance des lois physiques qui le régissent afin de pouvoir obtenir des prévisions fiables à plus ou moins court terme de son évolution. Nous étudions d'abord un algorithme de minimisation de fonctionnelles non quadratiques, puis nous proposons plusieurs méthodes rétrogrades ayant pour but la reconstitution de l'état initial d'un système, la quasi-réversibilité et le nudging rétrograde. Nous appliquons ensuite un algorithme variationnel classique à un modèle d'océan quasi-géostrophique barocline avant de le comparer à une nouvelle classe de méthodes, dites duales, qui permettent la prise en compte systématique de l'erreur modèle. De nombreux tests numériques menés sur un modèle océanique montrent que les méthodes duales permettent d'obtenir plus rapidement de meilleurs résultats et de réduire l'influence d'un terme d'erreur dans les équations du modèle.

Mots clés : Assimilation de données, Méthodes variationnelles, Océanographie, Contrôle optimal, Nudging, Quasi-réversibilité, Dualité.

Summary

Data assimilation consists in combining in an optimal way the observations of a system and the knowledge of the physical laws which govern it, in order to be able to obtain reliable forecasts of its evolution. We study initially a minimization algorithm of non necessarily quadratic functionals, and then we propose several backward methods for the reconstruction of the initial state of a system, the quasi-reversibility and the retrograde nudging. We then apply a traditional variational algorithm to an oceanic baroclinic quasi-geostrophic model before comparing it with a new class of methods, dual ones, which allow systematically to take into account the model error. Many numerical tests carried out on a oceanic model show that the dual methods make it possible to obtain more quickly better results and to reduce the influence of an error term in the model equations.

Keywords : Data assimilation, Variational methods, Oceanography, Optimal control, Nudging, Quasi-reversibility, Duality.