HAL
open science

Applications des méthodes multigrilles à l'assimilation de données en géophysique

Emilie Neveu

To cite this version:

Emilie Neveu. Applications des méthodes multigrilles à l'assimilation de données en géophysique. Mathématiques générales [math.GM]. Université de Grenoble, 2011. Français. NNT : 2011GRENM009 . tel-00574221v2

HAL Id: tel-00574221

https://theses.hal.science/tel-00574221v2

Submitted on 2 May 2012

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

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

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 août 2006

Présentée par

Emilie Neveu

Thèse dirigée par **François-Xavier Le Dimet**
et codirigée par **Laurent Debreu**

préparée au sein du **Laboratoire Jean Kuntzmann**
et de **l'Ecole Doctorale "Mathématiques, Sciences et Technologies de l'Information, Informatique"**

Application des méthodes multigrilles à l'assimilation variationnelle de données en géophysique

Thèse soutenue publiquement le **31 mars 2011**,
devant le jury composé de :

M. Eric Blayo
Professeur UJF (LJK), Président
M. Serge Gratton
Professeur ENSEEIHT (CERFACS), Rapporteur
M. Marc Bocquet
Directeur de Recherche École des Ponts ParisTech, Rapporteur
M. Rémy Baraille
Ingénieur SHOM, Examinateur
M. François-Xavier Le Dimet
Professeur Emérite UJF (LJK) , Directeur de thèse
M. Laurent Debreu
Chargé de recherche INRIA (LJK), Co-Directeur de thèse

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITE DE GRENOBLE

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 7 août 2006

présentée et soutenue publiquement par

Emilie NEVEU

le 31 mars 2011

Application des méthodes multigrilles à l'assimilation variationnelle de données en géophysique

---

Thèse dirigée par François-Xavier LE DIMET et Laurent DEBREU

JURY

M. Eric Blayo	Professeur UJF (LJK)	Président du jury
M. Serge Gratton	Professeur ENSEEIHT	Rapporteur
M. Marc Bocquet	Directeur de Recherche École des Ponts ParisTech	Rapporteur
M. Rémy Baraille	Ingénieur SHOM	Examinateur
M. François-Xavier Le Dimet	Professeur Emérite UJF (LJK)	Co-directeur
M. Laurent Debreu	Chargé de Rercherche INRIA (LJK)	Co-directeur

Thèse préparée au sein du Laboratoire Jean Kuntzmann (LJK, MOISE équipe projet INRIA) dans l'École Doctorale "Mathématiques, Sciences et Technologies de l'Information, Informatique" avec le soutien du Cluster Environnement de la Région Rhône-Alpes

REMERCIEMENTS

Je saisis l'occasion de ces quelques pages de remerciements pour dévoiler qui se cache derrière ce travail. Un peu moins de 200 pages de discours technique et pointu peuvent paraître dépourvu de toute humanité. Pourtant mes nombreux contacts et échanges ont contribué à mener à terme ce projet de recherche.

Il y a un peu plus de quatre ans, je revenais du Chili après 9 mois d'échange. Avec une seule envie, y retourner. Pourtant, j'ai finalement décidé de rester à Grenoble pour faire une thèse. Maintenant que je peux faire le bilan de ces trois, quatre années, je constate que l'apport a été autant professionnel que personnel. J'ai découvert que le changement, l'enrichissement, se trouvent partout et tout le temps, et que l'on peut voyager sur place.

Le voyage a été avant tout intérieur. Mon exploration m'a emmenée dans le pays des méthodes numériques en tout genre. Sans mes précieux guides de thèse, Laurent et François-Xavier, je me serai perdue. Vous avez su m'orienter, me conseiller, m'éclairer, m'accompagner, tout en me laissant explorer par moi-même. Je vous remercie de m'avoir montré le chemin et de m'avoir communiqué votre passion.
Je remercie également le jury et les rapporteurs Serge Gratton et Marc Bocquet pour leurs précieuses remarques avant et pendant la soutenance, pour leur disponibilité, et leur intérêt pour notre travail et enfin pour leur souci de faire de l'ultime journée un moment agréable.
Je veux remercier particulièrement l'équipe MOISE. Oui, toute l'équipe MOISE ! Des permanents qui ont toujours été disponibles et de bons conseils jusqu'à Imma et Anne qui m'ont bien simplifié la vie. J'ai beaucoup aimé faire partie de cette équipe.
Mon séjour de Grenoble a été perturbé par un passage de six mois au Cicese, à Ensenada. Je remercie Boris d'avoir gardé mes trésors, Julio Sheinbaum pour son accueil et pour les discussions océanographiques, et Jean et Nicolas pour m'avoir initié aux joies mexicaines. Cette courte période mariachesque m'a également donné deux amis exceptionnels. Michelle y Julio, pusieron ají en mi vida y su amor sincero todavía me ayuda en los dificiles momentos.
Ce cheminement scientifique a parfois été laborieux. J'ai su trouver refuge à Valence. J'en remercie mes élèves qui m'ont permis d'illustrer mes talents sur d'autres plans que la seule réflexion. J'espère qu'ils n'ont pas trop souffert de mes quelques expérimentations.
Lors de ces échappées magistrales, j'ai toujours été très bien accueillie par toute l'équipe pédagogique de l'Esisar, que ce soient le personnel administratif, les professeurs, les chercheurs ou les

thésards. Je ne peux tous les citer mais je ne peux pas ne pas citer Romain. Romain, toi qui m'a aidée quand personne d'autre ne l'a fait, tes conseils et ton soutien m'ont été essentiels. Je te dois nombre de Guinness pour ça !

Dans la vie de tous les jours de la Tour, j'ai vu aller et venir des quantités de gens. Il va sans dire que les quelques thésards qui m'ont accompagnée ont toujours été de précieux alliés : Carine, Irène, Claire, Ibrahim, Morgan, Damien, Pierre Antoine, Roland, Innocent, Manel, Aymen, Vincent, Samuel, Thomas, Brice, Bertrand. J'en oublie forcément. Les pauses cafés du septième (ciel ?). Les sourires de Béné. Les 15h de voiture avec Thomas. L'engouement montagnard de Laurence. Les déboires sentimentaux de l'équipe de proba/stat. Les milliers de tablettes de chocolat. La présence rassurante de Claire dans mon bureau. Les siestes de David. La douceur de vivre d'Adrien. La bonne humeur de Mad. L'énergie positive de Jérémie (et ses yeux bleus). Le soutien psychologique de Mélanie (toujours là en période de crise). La complicité de Gaëlle. La délicate générosité de Franck, et sa compagnie qui s'est révélée particulièrement bénéfique en fin de rédaction.
Et puis parce qu'il n'y a pas que les maths dans la vie, j'ai pu me rafraîchir les idées aux côtés de Xavier, Nia, Romain, les gros lourds, Julien, Valentine, Guillaume, Pédro, Yannis, avec un spécial big-up pour Tom ; parce qu'il faut toujours avoir un roux pour ami. Merci aussi à Bastien pour avoir cru en moi.
J'ai pu me rafraîchir la tête dans l'eau aussi. Parmi mes multiples activités "extra-scolaires", naviguer est ce qui m'a apporté le plus. Une communauté, un engagement, une passion. La possibilité d'évasion au contact de Den's, J-V, Bat, Claude, Pierre, Laurent, Cécile, Lucas, Jan, Cricri, Yako, Mimil, Antoine, Popop, Rémi, Juliette, Fabian ... et tous ceux qui gravitent autour de cette sphère, en particulier Alex, Pierre, Alexia.
Le voyage s'est effectué en compagnie des copains, de ceux qui resteront toujours à bord avec moi (sauf quand c'est moi qui nage), comme une deuxième famille : Nico-barrage et son humour inégalable, Lio et son côté nature, Osiris et ses sourires malicieux après avoir mis de la raclette dans les chaussures d'un autre, Bézel et sa cool-ouéch-ouéch-attitude, Guigui et sa bonne humeur, Séb la légende et ses blagues ardéchoises.
Et je remercie aussi tous ceux croisés au détour d'un chemin qui m'ont remis du baume au coeur rien qu'en échangeant quelques mots, merci à ma voisine dont le rire puissant a piétiné avec allégresse ma déprime rédactionnelle un soir de janvier.
Et puis voilà on arrive à la fin des remerciements, là où c'est le moment de remercier la familia : père, mère, soeurs, grands-parents, et tutti quanti... Vous avez toujours été là pour moi et vous m'avez beaucoup beaucoup beaucoup aidé !

Et parce que c'est peut être le seul livre que j'écrirai de ma vie, je le dédie à Régis, Francis, et Serge ; j'ai failli soutenir le premier avril.

TABLE DES MATIÈRES

Introduction **11**

I État de l'art **21**

1 L'assimilation de données dans tous ses états **23**
1.1 Petit guide de voyage dans le monde de l'assimilation de données 24
1.1.1 Notations pratiques 25
1.1.2 Population 26
1.1.3 Lieux communs 28
1.2 Itinéraire conseillé : 4 pages pour découvrir l'assimilation de données 29
1.2.1 Les méthodes variationnelles 30
1.2.2 Les méthodes séquentielles 31
1.2.3 Les défis à relever 33
1.3 Au cœur des méthodes variationnelles 33
1.3.1 Les équations d'Euler Lagrange 34
1.3.2 La formulation duale 35
1.3.3 La méthode 4D-variationnelle (4D-Var) 36
1.4 Conclusion 38

2 L'assimilation variationnelle de données : un système optimal complexe **39**
2.1 Résolution du système optimal 40
2.1.1 Le calcul du gradient 40
2.1.2 Les méthodes de minimisation 41
2.2 Les méthodes incrémentales 43
2.2.1 Le principe général 44
2.2.2 La méthode incrémentale vue par le prisme de l'optimisation 46
2.2.3 La méthode incrémentale réduite 47
2.3 Accélérer la convergence 49
2.3.1 Intérêt du préconditionnement 49
2.3.2 Influence de la matrice **B** 51

2.4 Conclusion . . . 52

3 Les méthodes multigrilles **53**
3.1 Les méthodes multigrilles pour la résolution de systèmes linéaires . . . 54
3.1.1 Les méthodes de relaxation . . . 55
3.1.2 La propriété de lissage . . . 55
3.1.3 Correction par grille grossière . . . 57
3.1.4 Les cycles des méthodes multigrilles . . . 58
3.2 Etude de la convergence . . . 59
3.2.1 L'ellipticité . . . 60
3.2.2 Les opérateurs de transfert . . . 61
3.2.3 Facteurs de convergence . . . 63
3.2.4 La propriété d'approximation . . . 64
3.2.5 Illustration . . . 65
3.2.6 Résumé . . . 67
3.3 Les systèmes non linéaires exposés . . . 67
3.3.1 Les méthodes de résolution classiques . . . 68
3.3.2 La méthode Newton-multigrille (N-MG) . . . 70
3.3.3 L'algorithme Full Approximation Scheme (FAS) . . . 71
3.3.4 Confrontation et hybridation . . . 74

II Les méthodes multigrilles pour l'assimilation de données **77**

4 Les méthodes multigrilles pour l'assimilation de données variationnelle **81**
4.1 L'algorithme multigrille adapté au contrôle optimal . . . 82
4.1.1 L'algorithme . . . 82
4.1.2 Rappel des conditions d'utilisation . . . 84
4.2 En ce qui concerne l'assimilation de données . . . 84
4.2.1 L'ellipticité du système . . . 84
4.2.2 La méthode d'optimisation . . . 85
4.2.3 Étude de la convergence . . . 86
4.3 Application à une équation d'advection linéaire . . . 87
4.3.1 Présentation de la configuration . . . 87
4.3.2 La propriété de lissage . . . 89
4.3.3 Étude de la convergence . . . 96
4.4 Conclusion . . . 105

5 L'extrapolation à la non linéarité **107**
5.1 Les algorithmes étudiés . . . 108
5.1.1 La méthode FAS . . . 109
5.1.2 La méthode Gauss Newton - multigrille (GN-MG) . . . 110
5.1.3 La méthode multi-incrémentale . . . 112
5.1.4 Les contraintes d'utilisation . . . 114
5.2 Spécificité des tests non linéaires . . . 115
5.2.1 Le modèle et les observations . . . 115
5.2.2 La fonction coût . . . 116

5.2.3 Les deux grilles . . . 116
5.2.4 Paramètres et configuration . . . 117
5.3 Les multigrilles éprouvées . . . 118
5.3.1 Les variables d'initialisation et les diagnostics utilisés . . . 118
5.3.2 Erreurs d'approximation . . . 119
5.3.3 Paramètres critiques . . . 122
5.3.4 Monogrille vs. Multirésolution . . . 125
5.3.5 Préconditionnement . . . 128
5.4 Conclusion . . . 129

6 Application à un modèle Shallow water 133
6.1 Introduction . . . 135
6.2 Variational data assimilation . . . 136
6.2.1 A complex control optimal system . . . 136
6.2.2 Incremental methods to solve the optimal system . . . 136
6.3 Multigrid methods . . . 138
6.3.1 A way to accelerate iterative methods . . . 138
6.3.2 Convergence properties . . . 139
6.4 The Multigrids applied to 4D-Var . . . 140
6.4.1 Non linear minimization . . . 140
6.4.2 Gauss-Newton Multigrid . . . 141
6.4.3 Convergence properties . . . 141
6.5 A 2D Shallow water model . . . 142
6.5.1 The Shallow water equations . . . 143
6.5.2 The multigrid configuration . . . 144
6.5.3 Experimental set-up . . . 144
6.5.4 Numerical results without preconditionning . . . 147
6.5.5 Preconditionned cost function . . . 150
6.6 Conclusion . . . 152

Conclusion 155

INTRODUCTION

Océan.
Masse d'eau occupant à peu près les deux tiers d'un monde
destiné à l'homme - lequel est dépourvu de branchies.
- Ambrose Bierce -

Les océans recouvrent environ 70,8 % de la surface du globe (on peut le constater sur la projection de Fuller de la Terre, Figure 1) et contiennent environ 97 % de l'eau disponible sur la Terre. Près de la moitié des eaux océaniques dépasse 3 000 m de profondeur ; le point le plus profond est la fosse des Mariannes, avec 11 020 m de profondeur [1].

Fig. 1 – Projection de Fuller de la Terre (ou dymaxion) : projection cartographique de la Terre sur la surface d'un polyèdre. Cette projection créée par Richard Buckminster Fuller, en 1946, a l'avantage de ne pas déformer, ni de favoriser le Nord ou toute autre région du globe.

Ces quelques chiffres suffisent pour comprendre le rôle essentiel de l'océan dans la vie de milliards de personnes et de créatures.

[1] Une partie des informations présentées, dans l'introduction, a été trouvée sur le site internet de l'Ifremer [33]

C'est ainsi que 50 % de la population mondiale vivaient en 1994 (revue Science du 14/11/97) à moins de 200 km des côtes. Pour la richesse et la diversité de ces ressources, l'océan trouve une grande importance dans l'activité humaine. L'Homme va chercher dans les mers son approvisionnement en ressources minérales, grâce aux gisements métalliques, aux énergies fossiles (Figure 2) ou encore aux diamants. Il y puise également son garde-manger en développant la pêche, l'aquaculture, et la cueillette des algues. Il lui doit donc, en partie, sa survie économique avec le commerce, le tourisme et la pêche. En 2005, 90% du trafic mondial de marchandises emprunte la mer.

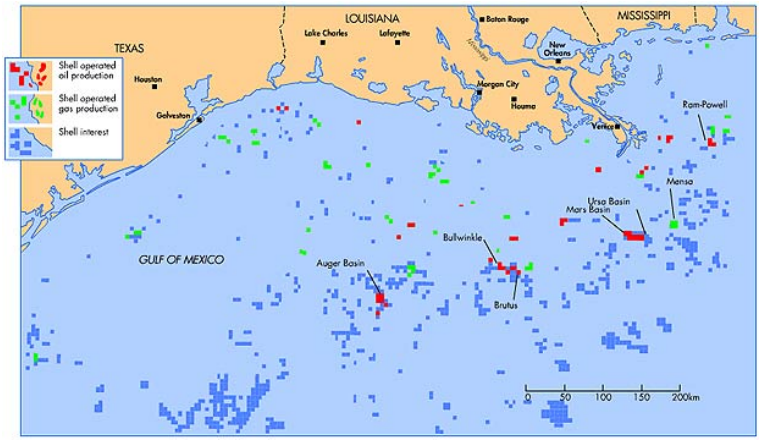

FIG. 2 – Champs d'hydrocarbures off-shore dans le Golfe du Mexique. En rouge : puits de pétrole ; en vert : puits de gaz naturel ; en bleu : prospection en cours

Le rôle des océans va bien au delà de la vie humaine, et influe également sur l'équilibre de l'écosystème terrestre dans son ensemble. Ils constituent, ce qui est souvent oublié, une réserve de diversité biologique équivalente ou supérieure à celle des forêts tropicales, notamment grâce aux coraux, dont les sites sont répertoriés sur la Figure 3.

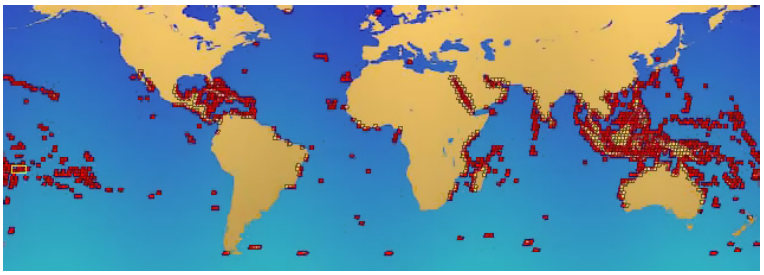

FIG. 3 – Répartition des récifs de coraux dans le monde. Contrairement à la croyance générale, les coraux ne sont pas réservés aux mers chaudes ; on en trouve par exemple au large de la Norvège

L'océan fait partie intégrante de tous les cycles importants de vie : le cycle de l'eau, de l'oxygène, du carbone, du soufre, et du phosphore. De plus, la majeure partie de la chaleur solaire reçue par notre planète est emmagasinée dans les mers, qui ont une capacité de stockage de la chaleur 1200 fois supérieure à celle de l'atmosphère.

Depuis toujours, l'étude des océans a été motivée par la curiosité et la recherche d'une meilleure compréhension de son fonctionnement. Dans un monde, où de plus en plus de ressources sont extraites des mers, et où l'enjeu des questions climatiques grandit de jour en jour, l'océanographie apparaît comme une science incontournable.
C'est également une science complexe et variée qui fait intervenir de nombreuses disciplines telles la biologie, la chimie, la géologie ou la physique. C'est à l'étude physique que nous nous intéressons et plus particulièrement à la dynamique des courants.
La prévision des courants permet, entre autres exemples, d'assurer la sécurité en mer des navigateurs, de prévoir l'impact des accidents comme les marées noires, et de mieux comprendre les liens avec l'atmosphère. Les connaissances actuelles, en matière de dynamique, sont relativement récentes. Nous allons faire un rapide aperçu de l'historique de ces connaissances qui nous viennent de l'exploration des océans et de l'étude des équations de la physique, pour ensuite être en mesure de présenter le cadre dans lequel ont été effectués ces travaux de thèse. Cela permet aussi de bien en mesurer les enjeux.

1 Ce qui se cache derrière l'étude des océans

Des premières explorations de l'océan aux observations satellites

La découverte des océans a été longue et difficile. Elle n'a débuté réellement qu'il y a quelques siècles avec le développement de la navigation et des premières explorations.
Pendant longtemps, les navires apportaient l'essentiel des connaissances de ce milieu et ces connaissances étaient éparses dans le temps et l'espace. De plus, les premières observations étaient limitées à la surface. C'est en 1840 que Sir James Clark Ross procèda au premier sondage moderne dans les mers profondes. Quelques dizaines d'années plus tard, John Murray et Charles Wyville lancèrent la première expédition appelée Challenger (1872-1876). Durant les trois ans du voyage, elle recueillit de nombreuses données sur la température, la salinité et la densité de l'eau de mer et donna une idée plus précise des fonds sous marins et des espèces y vivant.
L'expédition du Challenger fut suivie par de nombreuses autres missions scientifiques. Se généralisèrent alors les mesures in situ, effectuées, dans l'océan même, par le regroupement de plusieurs appareils de mesures au sein de plate-formes flottantes ou sur des bateaux lors de campagnes de mesure. On recueille ainsi, à l'aide de bathysondes, des informations précises sur le relief des fonds océaniques, la température, la salinité ou le taux d'oxygène en fonction de la profondeur. La vitesse et la direction des courants d'un point fixe sont collectées grâce aux courantomètres et les flotteurs dérivants permettent l'étude des masses d'eau profondes ou de surface sur une grande période.
Toutes ces mesures, bien que de plus en plus précises grâce aux perfectionnements des instruments (auparavant l'eau était prélevée dans un seau puis on mesurait sa température), restent très difficiles à mettre en oeuvre. Il est impossible d'effectuer des relevés prolongés partout, et les mesures effectuées sont inégalement réparties, puisque dépendantes des voies de navigation.

En 1960, une première révolution technologique a lieu. C'est l'année où fut lancé le premier satellite d'observation de l'atmosphère, TIRONS (Television InfraRed Observational Satellite). Ce satellite américain était à orbite polaire et donnait une image granulaire peu exploitable.
Depuis la fiabilité et la précision de ces satellites ont augmenté et sont maintenant suffisantes pour donner une image régulière quasi exhaustive de la planète. Ces satellites tournent sur une orbite quasi circulaire dont l'altitude est fixe. Un satellite couvre ainsi la totalité du globe en 24h

et donne une image haute résolution (1 km entre chaque pixel) d'une même région toutes les 12 heures.
Les satellites géostationnaires, quant à eux, offre un système permanent d'observations depuis l'espace. Situés à 35 800 km d'altitude, leur orbite est telle que le satellite reste dans la même position par rapport à la Terre. Ils sont au nombre de 5 et sont disposés en couronne à la verticale de l'équateur (GOES W, GOES E (américains), Meteosat (européen), INSAT (indien) et GMS (japonais)). La juxtaposition de leurs champs de vue recouvre la planète entière excepté les pôles et on a accès toutes les demi heures aux observations de bandes de latitudes situées entre 65° Nord et Sud. Leur inconvénient est dû à leur éloignement qui limite la qualité des images, la résolution obtenue est ainsi de 5 à 10 km.
L'immense apport des observations satellitaires réside dans la capacité à prendre du recul pour observer la planète dans son ensemble. Les images satellites font apparaître les structures dynamiques, tels les fronts et les tourbillons, mieux qu'aucune autre source d'information. Il est intéressant de savoir que grâce aux techniques d'observation spatiale, on peut mesurer la topographie de la mer (niveau des mers) avec une précision de l'ordre du mètre. Ajouté à cela, l'utilisation de mesures infrarouges pour les observations nocturnes et des instruments ultra sensibles qui permettent de "voir" à travers certains nuages, les satellites sont devenus essentiels pour la compréhension et la prévision des phénomènes météorologiques et océaniques.

Ces mesures satellites ne sont cependant pas parfaites. Imprécises de part la distance d'observation, et de part les erreurs instrumentales, ces données collectées sous forme de signaux nécessitent, de plus, un post-traitement afin de reconstituer les variables observées.
Enfin, l'ensemble de ces mesures ne donne qu'une vue instantanée de l'état de l'océan. Il faut lier ces vues les unes aux autres, trouver les états intermédiaires et ainsi comprendre la nature physique des mouvements observés. Pour cela, l'observation des océans est intrinsèquement liée à l'étude des phénomènes physiques et leur mise en équation.

La dynamique des fluides

Tout commença lorsque Archimède (-287,-212) cria "Eurêka" et énonça le célèbre théorème à son nom sur les corps plongés dans l'eau. C'est ainsi qu'apparaît la statique des fluides. Il faut néanmoins attendre le dix-septième siècle pour voir les connaissances en mécanique des fluides évoluer réellement et aboutir à l'étude de la dynamique.
Les équations de la dynamique des fluides proviennent des principes généraux de la physique : la conservation de la masse et de la quantité de mouvement ainsi que la conservation de la température (ou de l'énergie) et de la salinité. Elles permettent de connaître l'évolution de particules de fluide dans le temps et dans l'espace. Une particule fluide est un volume d'eau suffisamment petit pour que le fluide y ait des propriétés uniformes mais suffisamment grand pour que les quantités thermodynamiques comme la température y soient définies.
De nombreux chercheurs, dont L. Euler (1707-1783), D Bernoulli (1700-1782), H. Navier (1785-1836), G. Stokes (1819-1904) et J-L. Poiseuille (1799-1869), apportèrent leur pierre à l'édifice. Les équations de Navier-Stokes, bien connues des géophysiciens, déterminent le mouvement des fluides visqueux. Elles furent introduites pour la première fois en 1822 par Navier dans le cas de la viscosité constante.
Ce fut alors le début d'une nouvelle ère. Les études sur la dynamique des fluides se multiplièrent et prirent de nombreuses directions avec, par exemple, l'étude de la stabilité des écoulements (H. Helmoltz (1868), Kelvin (1910)), le transport de la chaleur par convection (W. Prout (1834),

Rumford (1870), A. Oberbeck (1879), J. Boussinesq (1903), H. Bénard (1900), Lord Rayleigh (1916)), ou la turbulence (O. Reynolds (1883), L. Prandtl (1920), A. Kolmogorov (1940)), qui reste l'un des grands axes de recherche, de la physique contemporaine.

La dynamique des fluides étudie les écoulements de fluides tels les liquides ou les gaz. Elle peut aussi être étendue à des écoulements "solides" comme les glaces ou le manteau terrestre. Elle a donc de multiples applications dans l'aéronautique avec l'aérodynamique, la médecine et, par exemple, l'étude des écoulements sanguins (hémodynamique), et également la météorologie, la climatologie, l'hydrologie et l'océanographie physique.
Les équations de la dynamique océanique sont les équations de Navier-Stokes, adaptées au "fluide" océan. C'est-à-dire qu'on se place dans le cas particulier où le fluide étudié est l'eau, soumis à la force d'inertie de la Terre en rotation sur elle-même.

La révolution des ordinateurs

Depuis presque deux siècles, les lois fondamentales de la dynamique des fluides sont connues. Ces équations différentielles décrivent des relations entre une ou plusieurs fonctions inconnues et leur dérivées qui peuvent être des dérivées partielles. Il n'existe que rarement de solutions analytiques à ces équations. Si on ne cherche plus à connaître la solution en tout point de l'espace et en tout temps, mais seulement sur quelques points, on peut résoudre ces équations numériquement. Résoudre numériquement une équation différentielle consiste à approcher, le plus précisément possible, la solution en un certain nombre de points.
Le choix des points de calculs consiste à créer ce qu'on appelle le maillage. On construit ensuite un système liant les valeurs nodales (les valeurs de la solution numérique aux points de calcul choisis) entre elles, via l'approximation numérique des dérivées. Les modèles diffèrent aussi par le choix du type d'approximation. Les modèles de prévision océanographiques sont presque tous basés sur une approximation aux différences finies. C'est-à-dire qu'on utilise un développement de Taylor d'ordre plus ou moins élevé pour approcher chaque dérivée. Mais il existe également des schémas aux volumes finis et aux éléments finis. L'idée maîtresse sur laquelle repose ces approches est que, la solution approchée (celle obtenue aux points de calcul choisis) tend vers la solution exacte du problème différentiel traité, l'approchant d'autant mieux que le nombre de valeurs nodales résolues est grand.

Il y a 65 ans maintenant, que le premier ordinateur, entièrement électronique, a été inventé. Une des premières utilisations de ces machines a été de résoudre les équations différentielles de la dynamique des fluides ou, plus exactement, d'approcher la solution, de manière beaucoup plus rapide et plus sûre qu'un humain, permettant alors des progrès considérables en simulation numérique.

Il va sans dire que la construction des systèmes n'est pas anodine. Un bon modèle dépend de la qualité de la physique résolue, mais également de la qualité des approximations numériques.
Nous mettons en évidence d'une part les erreurs du modèle dues à une simplification ou à une méconnaissance de la physique, et d'autre part les erreurs de discrétisation. Ces erreurs ne sont, cependant, pas un obstacle à l'utilisation des modèles. En effet, le modèle est conçu pour répondre à une question. Selon les objectifs, un même système peut être modélisé différemment. Si on sait que la topographie n'influe pas, et que les mouvements du fluide sont principalement horizontaux, le modèle va pouvoir prendre en compte cette simplification. Si obtenir un modèle

parfait est impossible, ce n'est pas non plus un but en soi.
À méditer cette fable de Borges :

Dans cet Empire, l'Art de la Cartographie avait atteint une telle perfection que la carte d'une seule province occupait toute une ville, et que la carte de tout l'Empire occupait toute une province. Avec le temps, ces cartes démesurées ne furent plus suffisantes et les collèges de cartographes firent une carte de l'Empire qui avait la taille de l'Empire et qui coïncidait point par point avec lui. Moins attachées à l'étude de la cartographie, les générations suivantes comprirent que cette carte dilatée était inutile et, non sans impiété, ils l'abandonnèrent aux inclémences du soleil et au froid de l'hiver.
Suárez Miranda : *Viajes de Varones prudentes*, (1658)
imaginé par J.-L. Borges,
Histoire universelle de l'infamie/Histoire de l'éternité.

Les modèles de simulation océanographique

Après une marée noire, des modèles de simulation permettent de prévoir l'impact des fuites de pétrole et leur destination sur les côtes. Ces modèles ont un but de prévision. Leur domaine d'intérêt est relativement restreint. Il s'agit de modèle régionaux. Mais de nombreux autres modèles existent dont la complexité varie selon leur utilité.
Il est difficile de les classifier mais l'on peut différencier deux types d'études selon que l'on veut prévoir de manière déterministe ou de manière statistique. Les études statistiques sont généralement basées sur l'intégration de plusieurs modèles dont on estime les erreurs de prévision pour utiliser une moyenne pondérée des différentes prévisions.
Les études déterministes sont basées sur un unique modèle. Il peut s'agir de modèles globaux, où toute la superficie terrestre est modélisée, de modèles régionaux ou encore de modèles côtiers dont les équations physiques sont adaptées aux ruptures de topographie et aux vagues.
Les simulations océanographiques peuvent être enrichies par des modèles biochimiques, des modèles de glace flottante ou encore couplées à des modèles atmosphériques, ce qui nous donne des représentations plus précises de la physique. Mais il est encore très souvent utilisé des modèles "jouets", très simples, pour tester une méthode numérique ou parce qu'une ébauche de la physique est suffisante.

2 Cadre de la thèse et objectifs

Depuis à peine deux siècles, les instruments d'observation de la Terre sont apparus et ont augmenté considérablement en nombre, qualité et couverture ces 60 dernières années. Parallèlement à ce développement, les équations de la dynamique des fluides ont vu le jour. Elles ont permis la création de systèmes d'équations simulant l'état de l'océan ou de l'atmosphère. Ces systèmes se sont diversifiés et complexifiés, avec la prise en compte de plus en plus de détails de configurations et de mécanismes physiques, biologiques ou chimiques.

Les modèles de prévisions géophysiques calculent les états successifs du fluide à partir d'un état initial qui amorce le calcul. Le choix de cet état initial est fondamental. En effet, la nature chaotique des équations, non linéaires, fait qu'une petite perturbation locale peut entraîner une modification complète de l'état du fluide au bout de, quelques semaines pour l'océan, quelques jours pour l'atmosphère. C'est ce que l'on appelle communément " l'effet papillon ". L'état initial de la prévision est donc un paramètre sensible.
On peut illustrer ce phénomène en lançant deux simulations amorçées par des états initiaux très légèrement différents. C'est ce qu'a fait Lorenz [42] dans les années 1960 sur un modèle simplifié de la convection atmosphérique. Il a prouvé ainsi qu'une infime variation ou imprécision dans les conditions initiales menait à des prévisions météorologiques complètement opposées au bout d'un certain temps. Par la suite, les résultats de Lorenz seront remis partiellement en question. Par exemple, les travaux de Robert [52] montrent que le système évolue différemment suivant le nombre de degrés de liberté que l'on prend en compte. Bien que les trajectoires des particules fluides soient chaotiques et imprévisibles à cause de la turbulence à petite échelle, on peut prévoir le champ des vitesses macroscopiques du fluide (sauf cas exceptionnel).
Le grand changement réside dans le pronostic de prévision. L'erreur de prévision ne double pas à chaque fois que le temps augmente d'une valeur donnée, comme le prévoyait Lorenz, mais elle croît de façon linéaire. La prévision à long terme (au delà de deux semaines pour l'atmosphère) devient possible si on améliore la connaissance de l'état initial de l'atmosphère ou de l'océan. Améliorer cet état est d'autant plus nécessaire pour la prévision de phénomènes extrêmes tels les orages ou les tempêtes en météorologie et pour une prévision plus fiable.

Il ne paraît pas illusoire de réussir à déterminer un état initial d'après les prévisions numériques de la veille et les mesures collectées un peu partout au cours du même laps de temps. Mais comment construire cet état initial ? Comment combiner au mieux des informations aussi hétérogènes que les mesures et le modèle ?
C'est pour répondre à ces questions qu'ont été développées les méthodes d'assimilation de données. Elles consistent à coupler toutes les informations en notre possession, que ce soit la dynamique du modèle ou les observations passées, afin d'estimer l'état du fluide avant de lancer le calcul de prévision.
Parmi ces méthodes, l'assimilation de données variationnelle est utilisée depuis les années 90 en situation opérationnelle. De plus en plus de centres de prévisions météorologiques la choisissent : Météo France, European Center for Medium-range Weather Forecasts (Europe), Center for Ocean-Atmosphere Prediction Studies (Floride), National Center for Earth Observation (UK), MetOffice (UK), Japan Meteorological Agency (Japon) et Environment Canada ; ainsi que des modèles opérationnels océanographiques (ROMS, NEMO). Elle est basée sur la minimisation d'une fonctionnelle mesurant l'écart du modèle aux observations.

L'efficacité de l'assimilation variationnelle n'est plus à prouver mais sa mise en œuvre demeure difficile. En effet, elle demande de très grandes ressources de calcul. De plus, pour des modèles non linéaires, les fonctionnelles non quadratiques sont difficiles à minimiser de par leur non-convexité et l'existence éventuelle de plusieurs minimum locaux. Il est donc intéressant pour tirer profit des bons résultats de l'assimilation de chercher à optimiser le calcul.
Pour ces raisons, plusieurs stratégies ont été mises en place. Les problèmes de minimisation sont le plus souvent contournés en résolvant une suite de problèmes quadratiques (approches incrémentales) qui peuvent être alors résolues avec des méthodes d'optimisation efficaces. La convergence de ces méthodes n'est cependant pas assurée et requiert une très bonne connaissance d'une approximation de l'état optimal. La seconde simplification consiste à diminuer la résolution du modèle afin de calculer ces incréments. On s'attache ici à la diminution du coût de calcul. Les deux approches peuvent être couplées au sein de la méthode nommée multi-incrémentale réduite (Courtier [11], Veersé et Thépaut [63]).
Pourtant si l'on veut pouvoir prendre en compte l'information à toujours plus haute résolution des modèles et des observations, l'assimilation de données doit être résolue *in fine* à haute résolution.

Ces travaux de thèse ont pour objectif de continuer la réflexion déjà commencée par Ehouarn Simon, Laurent Debreu et Eric Blayo ([55] , [14]) sur l'assimilation de données dans le cadre de modèles multirésolutions ; et d'étudier une alternative aux méthodes multi-incrémentales. Nous souhaitons aborder ici le problème sous un autre angle : utiliser plusieurs maillages selon une méthode de résolution multigrille. Les méthodes multigrilles sont basées sur des méthodes de relaxation dont elles améliorent les propriétés de convergence en calculant une correction à plus basse résolution. Elles ont été appliquées sur de nombreux problèmes dont la résolution de système optimal avec ou sans contraintes (Ta'asan [59]). Nous étudions ici leur application à des fonctionnelles typiques de l'assimilation de données géophysiques.

3 Organisation du manuscrit

Nous abordons dans les deux premiers chapitres la problématique de l'assimilation de données, en la présentant d'un point de vue général pour se concentrer ensuite sur l'assimilation de données variationnelle, et les difficultés rencontrées pour sa mise en œuvre. Nous présentons alors le fonctionnement des méthodes multigrilles au chapitre trois.
Les méthodes multigrilles sont appliquées, aux chapitres suivants, à la résolution du système optimal de l'assimilation de données. Nous étudions tout d'abord un modèle linéaire à une dimension et l'influence des paramètres d'assimilation sur la résolution. Le cinquième chapitre élargit l'étude aux modèles non linéaires, c'est à dire une équation de Burgers à une dimension, et s'intéresse à la robustesse des différents algorithmes multigrilles. Enfin le dernier chapitre est l'occasion d'évoluer vers les modèles de prévisions opérationnelles. Les méthodes multigrilles sont testées sur un modèle Shallow-water à deux dimensions. Nous mettons en évidence les limites et surtout les perfectionnements possibles pour que la résolution multigrille demeure plus efficace et plus rapide qu'une méthode monogrille.

Première partie

État de l'art

1

L'ASSIMILATION DE DONNÉES DANS TOUS SES ÉTATS

Sommaire

1.1 Petit guide de voyage dans le monde de l'assimilation de données . 24
1.1.1 Notations pratiques 25
1.1.2 Population 26
1.1.3 Lieux communs 28
1.2 Itinéraire conseillé : 4 pages pour découvrir l'assimilation de données 29
1.2.1 Les méthodes variationnelles 30
1.2.2 Les méthodes séquentielles 31
1.2.3 Les défis à relever 33
1.3 Au cœur des méthodes variationnelles 33
1.3.1 Les équations d'Euler Lagrange 34
1.3.2 La formulation duale 35
1.3.3 La méthode 4D-variationnelle (4D-Var) 36
1.4 Conclusion 38

Avec l'augmentation du nombre et de la fiabilité des observations de l'océan, et avec la précision toujours plus grande des modèles numériques, a commencé la création de systèmes de prévisions océanographiques opérationnelles. Par analogie avec la météorologie, un système d'océanographie opérationnelle est constitué d'une étape d'analyse entre les observations mesurées et les états d'un modèle. Puis, le modèle évolue dans le temps pour faire une prévision des conditions océaniques pour les jours qui suivent. L'assimilation de données est ce qui permet d'analyser les données. Elle utilise toute l'information en possession, c'est-à-dire les mesures et les sorties du modèle, pour construire un état à partir duquel lancer la prévision.
Deux types de méthodes ont été introduites par Sasaki [53] en 1955, et Kalman en 1960 [35] : les méthodes séquentielles, appliquées à la météorologie dès 1965 [34] et les méthodes variationnelles, appliquées pour la première fois, dans les années 90, aux modèles réalistes atmosphériques

[60] et océanographiques [47], puis utilisées dans les centres de prévisions opérationnelles de l'atmosphère, European Center for Medium-range Weather Forecasts depuis 1997 et Météo France depuis 2000.
Ces méthodes ont pour nom générique "méthodes d'assimilation de données".
Elles consistent à coupler toutes les informations en notre possession, que ce soit la dynamique du modèle ou les observations passées, afin d'estimer l'état du fluide avant de lancer le calcul de prévision. En considérant que les mesures et le modèle ne sont pas parfaits et contiennent donc des erreurs, il s'agit de calculer ces différentes erreurs en interpolant la dynamique et les observations.
Après assimilation, on obtient alors un état optimal. La figure 1.1 illustre un processus d'assimilation qui calcule cet état optimal ; les différentes méthodes représentées seront explicitées un peu plus loin. La prévision peut alors être lancée avec pour état initial, l'état optimal trouvé après assimilation.

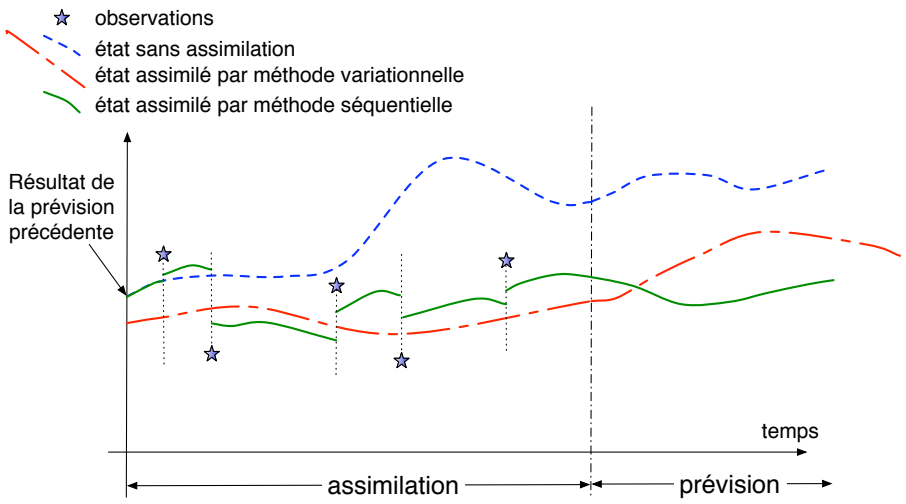

FIG. 1.1 – Un processus d'assimilation de données

Certaines méthodes d'assimilation de données permettent également de calculer à quels paramètres entrant en jeu, à quelles erreurs et à quelles observations, le modèle est le plus sensible. On peut ainsi mieux comprendre les phénomènes simulés. Cependant ces analyses de sensibilité ne sont pas encore opérationnelles pour les modèles géophysiques car très coûteuses.

Nous allons présenter, dans ce chapitre, une vue générale de l'ensemble des méthodes d'assimilation et se focaliser sur les méthodes dites variationnelles que nous utilisons dans cette thèse. Mais avant tout, il est préférable d'introduire les notions et notations utilisées par la suite.

1.1 Petit guide de voyage dans le monde de l'assimilation de données

Comme tout voyage en terre inconnue, entrer dans le monde de l'assimilation de données peut être déroutant et il vaut mieux avoir en sa possession un petit guide, qui explique les moeurs et coutumes locales et explicite deux ou trois phrases de vocabulaire de la langue parlée dans ce

monde imaginaire. Ce qui suit est un concentré du guide réalisé par Ide *et al.* [32] en 1997. Il unifie les différentes notations, utilisées jusqu'alors, par les différents utilisateurs des méthodes d'assimilation.

1.1.1 Notations pratiques

Tout d'abord il faut savoir reconnaître à qui on a affaire. Pour cela, on utilise la typographie des notations. Avec un peu d'entraînement, et à l'aide du Tableau 1.1, nous pourrons reconnaître une valeur scalaire en voyant une minuscule en italique.

Typographie	Identification
a	valeur scalaire
A	opérateur non linéaire
$\mathbf{a}$	vecteur
$\mathbf{A}$	opérateur linéaire ou matrice

TAB. 1.1 – Significations de la typographie utilisée

Ensuite, il faut faire attention, nous serons amenés à rencontrer assez souvent ces "êtres" portant un exposant sur la tête. Cet exposant apporte une information supplémentaire. Rappelons avant tout que pour faire de l'assimilation de données, il faut avoir un modèle sous la main. Ce modèle prédit un état du système en partant d'un état initial. L'état du système est dans notre cas, l'état des variables calculées par le modèle géophysique. Tout ce qui portera un exposant ^{t} (signifiant *true*) représentera l'état vrai (la transposition de la réalité, de la vraie solution, dans l'espace des variables du modèle).
L'assimilation de données consiste à "caler" les paramètres et les erreurs du modèle et des observations pour que l'état prédit (ou la prévision) approche *au mieux* les données et la physique du modèle. La connaissance de l'état prédit est alors *optimale* et on peut l'utiliser comme état initial pour lancer une prévision. Parmi les paramètres du modèle qui sont susceptibles de varier pendant l'assimilation, il y a la condition initiale du modèle. On notera avec un exposant ^{b} les variables qui se rapportent à l'état initial estimé avant toute assimilation, cet état est aussi noté ébauche ou *background*. Enfin l'exposant ^{a} sera utilisé pour les variables correspondant à l'état trouvé après assimilation, appelé également état analysé. Cet exposant, contrairement à l'exposant ^{b} peut être utilisé pour d'autres variables que les variables initiales.

Exposant	Signification
t	état vrai
b	ébauche ou premier état initial du modèle (avant assimilation)
a	état analysé, état prédit après assimilation
f	état transcrit sur la grille haute résolution (*fine grid*)
c	état transcrit sur la grille basse résolution (*coarse grid*)
h	état transcrit sur une grille de résolution h en espace

TAB. 1.2 – Significations de l'exposant utilisé

À l'exception de ce chapitre qui présente l'assimilation de données dans un cadre général, il sera utilisé plus d'une grille de discrétisation pour résoudre l'assimilation de données. Nous distinguons également les états selon la grille sur laquelle ils sont transcrits en utilisant les exposants ^{f}, ^{c} et ^{h}.
On trouve un récapitulatif des différents exposants dans le tableau 1.2.

1.1.2 Population

Le vecteur d'état x

Le vecteur d'état représente l'ensemble des variables pronostiques du modèle. Dans des modèles complexes tels les modèles océans, le vecteur d'état, noté génériquement $\mathbf{x}$, est un assemblage de plusieurs variables, par exemple la vitesse du courant, la température et la salinité. On note n sa dimension, $\mathbf{x} \in \mathbb{R}^n$ et $\mathbf{x}_i$ décrit l'état de l'océan à l'instant t_i.

Le vecteur de contrôle u

Le vecteur de contrôle représente l'ensemble des paramètres du modèle que l'on cherche à estimer. Il peut inclure les valeurs de coefficients de viscosité ou les conditions aux limites. Cependant sa dimension reste plus petite que celle du vecteur d'état. Nous supposons dans toute la suite que l'unique paramètre imparfait du modèle est la condition initiale, que l'on note $\mathbf{x}_0$:

$$\mathbf{u} = \mathbf{x}_0$$

Cette hypothèse est un souci de simplification, l'état initial étant, dans une optique de prévision, le paramètre le plus sensible des modèles géophysiques.

Le modèle M

Le modèle de prévision peut être différent selon le problème étudié. Nous allons englober tous les modèles possibles, implicites et explicites en temps, sous la notation générique :

$$F(\mathbf{x}, \mathbf{u}) = 0$$

Dans le cas de modèles explicites en temps, l'état $\mathbf{x}_{i+1}$ à l'instant t_{i+1} peut être décrit en fonction de l'état $\mathbf{x}_i$ de l'instant t_i si l'on considère que $M_{i,i+1}$ désigne l'évolution temporelle du système de l'instant t_i, jusqu'à l'instant t_{i+1}. On utilise alors le système d'équations suivant :

$$\mathbf{x}_{i+1} = M_{i,i+1}(\mathbf{x}_i)$$

Les erreurs du modèle

Si l'on tient compte des erreurs que l'on fait en modélisant, l'état prédit par l'équation précédente n'est pas exact. Pour trouver la solution exacte, il faut calculer, pour un modèle explicite :

$$\mathbf{x}_{i+1}^t = M_{i,i+1}(\mathbf{x}_i^t) + \boldsymbol{\eta}_i$$

Sans informations supplémentaires sur la nature de l'erreur du modèle, $\boldsymbol{\eta}_i$ est modélisé par une perturbation stochastique, c'est-à-dire un bruit dont on estime les paramètres statistiques. Ainsi, η est un processus stochastique de moyenne :

$$\bar{\boldsymbol{\eta}}_i = E[\boldsymbol{\eta}_i]$$

où $E[.]$ représente l'espérance mathématique, et de matrice de covariance :

$$\mathbf{Q}_i = E[(\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}_i)(\boldsymbol{\eta}_i - \bar{\boldsymbol{\eta}}_i)^T]$$

Ces perturbations représentent tous les processus non modélisés. Ils englobent également les processus non résolus par le modèle car d'échelles plus petites que l'échelle de discrétisation du modèle.
Si l'on suppose que le modèle, sa discrétisation, ainsi que les données extérieures paramétrant le modèle (vents et flux de chaleur, par exemple) sont de bonnes approximations de la réalité, les erreurs du modèle sont petites.

Les observations $\mathbf{y}$

Le vecteur d'observation $\mathbf{y}$ rassemble les mesures disponibles pendant l'intervalle de temps où on applique l'assimilation de données. Il peut s'agir de mesures in situ, comme les bouées, ou d'informations provenant des satellites comme l'élévation de la hauteur d'eau. Si la taille du vecteur d'état dépend de la taille de discrétisation du modèle, la taille du vecteur d'observations est en général limitée par le nombre d'observations en possession. En raison de problèmes de stockage et de traitement de l'information, la haute résolution des images satellites n'est pas encore tirée à profit. C'est pourquoi la taille de $\mathbf{y}$ est plus petite que celle de $\mathbf{x}$.

L'opérateur d'observation discret H

Nous savons que les observations de l'océan ne sont pas continues dans le temps et l'espace. Leur localisation spatio-temporelle ne correspond pas nécessairement à la localisation spatio-temporelle à laquelle on calcule les différentes variables du modèle. De plus, les variables observées ne sont pas forcément de la même nature que les variables simulées. Par exemple, il peut s'agir de mesures de rayonnement infrarouge issues des satellites.

Il est donc essentiel de définir un opérateur d'observation H qui traduit les variables $\mathbf{x}$ de l'espace du modèle dans l'espace des observations, pour pouvoir les comparer.

On suppose que l'on dispose, à chaque temps t_i, $i = 1, 2, ...$, d'un nombre p_i d'observations : $\mathbf{y}_{t_i} \in \mathbb{R}^{p_i}$. L'opérateur d'observation est ainsi défini :

$$\mathbf{y}_{t_i} = H_{t_i}(\mathbf{x}_i^t) + \boldsymbol{\varepsilon}_{t_i}$$

Il peut être non linéaire et inclure une dépendance temporelle. $\boldsymbol{\varepsilon}_{t_i}$ représente l'erreur d'observation. Ces erreurs sont d'une part des erreurs de mesure, provenant de l'imprécision ou du dérèglement des appareils de mesure, et qui sont indépendantes des erreurs du modèle et d'autre part des erreurs dites de représentativité correspondant aux erreurs engendrées par la représentation des mesures dans l'espace discret du modèle.

Les erreurs d'observations

Les erreurs d'observations sont considérées être un processus stochastique, dont on estime la moyenne :

$$\bar{\boldsymbol{\varepsilon}}_{t_i} = E[\boldsymbol{\varepsilon}_{t_i}]$$

et la matrice de covariance

$$\mathbf{R}_i = E[(\boldsymbol{\varepsilon}_{t_i} - \bar{\boldsymbol{\varepsilon}}_{t_i})(\boldsymbol{\varepsilon}_{t_i} - \bar{\boldsymbol{\varepsilon}}_{t_i})^T]$$

L'ébauche $\mathbf{x}^b$

L'ébauche est une estimation de la condition initiale du modèle. Typiquement il s'agit de l'état initial de la période d'assimilation du modèle. On le connaît a priori. De la même façon que l'on a défini une erreur du modèle et une erreur d'observation, on représente $\boldsymbol{\varepsilon}^b$ comme étant l'erreur d'ébauche :

$$\boldsymbol{\varepsilon}^b = \mathbf{x}^b - \mathbf{x}_0^t$$

caractérisée, elle aussi, par sa moyenne et sa matrice de covariance notée $\mathbf{B}$.

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

Les erreurs de prévision

L'erreur de prévision que l'on note $\mathbf{x}_i - \mathbf{x}_i^t$ a pour matrice de covariance :

$$\mathbf{P}_i = E[(\mathbf{x}_i - \mathbf{x}_i^t).(\mathbf{x}_i - \mathbf{x}_i^t)^T]$$

Il s'agit de l'écart entre la solution du modèle et la solution exacte si $\mathbf{x}_i$ est solution du modèle.

Les erreurs d'analyse

L'erreur d'analyse, notée $\mathbf{x}_i^a - \mathbf{x}_i^t$ et de matrice de covariance

$$\mathbf{P}_i^a = E[(\mathbf{x}_i^a - \mathbf{x}_i^t).(\mathbf{x}_i^a - \mathbf{x}_i^t)^T],$$

représente l'écart de la solution exacte avec la solution du modèle après la période d'assimilation des données.

1.1.3 Lieux communs

Après cette rapide présentation d'ordre pratique, nous constatons d'ores et déjà qu'il ne s'agit pas uniquement de modèles et d'observations mais également de statistiques d'erreur. Ce monde est peuplé de nombreuses variables difficiles à estimer, mais qu'il est pourtant nécessaire de connaître pour saisir toute la subtilité de ce voyage.
Pour simplifier les méthodes d'assimilation de données, nous allons donc considérer que les réductions ou simplifications suivantes, souvent faites, sont vraies.

- Erreur modèle :
  Nous supposons que le modèle est une bonne approximation de la réalité et donc que ces erreurs sont suffisamment petites pour être négligées : $\boldsymbol{\eta}_i = 0, \forall i$

– Erreur d'observation :
Elle représente les statistiques des erreurs de mesures et des erreurs de représentativité. Les premières sont dues aux instruments de mesures. Pour un satellite, l'instrument de mesure est toujours le même. Les mêmes erreurs de mesure se répètent donc dans l'espace, voire dans le temps. Elles peuvent donc être fortement corrélées entre elles. Les erreurs de représentativité sont, quant à elles, engendrées par le modèle et la discrétisation. Les erreurs d'interpolation sont corrélées lorsque les observations sont denses par rapport à la résolution du modèle. Enfin les erreurs du modèle, sont propagées par le schéma numérique discret et sont elles aussi corrélées.
Pourtant, en pratique, nous supposons que les erreurs ne sont pas corrélées spatialement et temporellement, qu'elles sont non biaisées ($\bar{\boldsymbol{\varepsilon}} = 0$). Ceci induit l'utilisation d'une matrice de covariances d'erreurs aux observations $\mathbf{R}$ diagonale. Cette hypothèse est faite parce que l'information statistique sur ces erreurs est peu connue, mais également parce que prendre une $\mathbf{R}$ non diagonale rend la minimisation difficile.

1.2 Itinéraire conseillé : 4 pages pour découvrir l'assimilation de données

Maintenant que nous avons présenté les différents êtres susceptibles d'être rencontrés dans ce pays de l'assimilation de données, nous pouvons décrire les différentes activités et régions à découvrir (ou redécouvrir). Dans ce monde, de nombreuses méthodes d'assimilation de données cohabitent malgré leurs différences. Ce paragraphe, loin d'avoir la prétention d'être un cours sur ces différentes méthodes, est plutôt écrit dans le but de passer en revue, d'un rapide coup d'oeil, les différentes "espèces" et de donner une idée de l'étendue des possibles. Pour plus de détails, un lecteur curieux pourra se reporter aux références bibliographiques.

Nous nous limitons à des modèles présentés sous la forme discrète et explicite mais il faut savoir que les développements qui suivent peuvent être appliqués de la même façon à des modèles continus et/ou implicites en temps. Dans la suite, nous considérons un modèle, qui peut être non linéaire, défini sur un domaine $\Omega \times [0, T]$ avec Ω le domaine spatial et T le temps d'intégration. Nous cherchons la solution $\mathbf{x}^t$:

$$\begin{cases} \mathbf{x}_{i+1} = M_{i,i+1}(\mathbf{x}_i) \\ \\ \mathbf{x}_0 = \mathbf{x}^b + \boldsymbol{\varepsilon}^b. \end{cases}$$

Les conditions aux bords du domaine sont comprises dans le modèle et nous considérons qu'elles sont connues exactement.
L'erreur aux observations est notée $\boldsymbol{\varepsilon}_{t_i} = H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}$.

Comme présenté plus haut, nous supposons que l'unique paramètre imparfait du modèle est la condition initiale, $\mathbf{x}_0$. Ainsi, nous cherchons l'état initial pour lequel les erreurs d'observations $\boldsymbol{\varepsilon}$ et l'erreur de l'ébauche $\boldsymbol{\varepsilon}^b$ seront les plus petites possibles. Ce qui va nous permettre d'obtenir une solution $\mathbf{x}$, proche des observations de la période d'assimilation et proche de l'ébauche. On suppose que la solution optimale trouvée sera la solution vraie : $\mathbf{x}^a = \mathbf{x}^t$.

Plusieurs méthodes existent pour trouver cet état initial. L'usage fait qu'on les regroupe dans deux catégories, les méthodes variationnelles et les méthodes séquentielles. Les premières sont

basées sur la théorie du contrôle optimal et les deuxièmes sur la théorie de l'estimation statistique optimale. Dans cette section, pour chaque méthode, nous décrivons le principe, les avantages et les inconvénients et enfin les défis qui restent à relever.

1.2.1 Les méthodes variationnelles

Utiliser une méthode variationnelle consiste à définir une fonction coût à minimiser. Cette fonction est une moyenne pondérée aux moindres carrés des différentes erreurs précédentes.
Plus particulièrement, on cherche l'état initial optimal $\mathbf{x}_0^a$ qui minimise la fonction $\mathcal{J}$ suivante :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{1.1}$$

N étant le nombre de dates d'observations.

Cette fonction peut s'écrire également :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}\boldsymbol{\varepsilon}_{t_i}^T\mathbf{R}_i^{-1}\boldsymbol{\varepsilon}_{t_i} + \frac{1}{2}[\boldsymbol{\varepsilon}^b]^T\mathbf{B}^{-1}[\boldsymbol{\varepsilon}^b]$$

Dans le cas où l'opérateur d'observation et le modèle sont linéaires, la méthode variationnelle est en fait équivalente à l'approche statistique de maximum de vraisemblance. En effet, on s'aperçoit que l'on effectue une moyenne pondérée des différentes erreurs et pour chaque type d'erreur, le poids utilisé est l'inverse de la matrice de covariance correspondante. Ceci est logique car cette matrice quantifie l'importance des erreurs. Plus les erreurs d'une donnée sont supposées grandes, moins on aura confiance en elle.

Pour minimiser une fonction, il existe plusieurs méthodes. Elles dérivent toutes du calcul variationnel, d'où le nom donné aux méthodes d'assimilation de données basées sur la minimisation d'une fonction coût. Plus de détails seront donnés à la section suivante.
Après résolution, on obtient les paramètres optimum du système ; qui, dans notre cas, sont réduits à l'état initial. La solution est déduite après intégration du système avec ces nouveaux paramètres, elle a donc une cohérence dynamique en temps et en espace. Si le modèle utilisé est continu, nous obtenons une solution continue dans le temps telle qu'on peut le voir sur la figure 1.1, où la solution donnée par une méthode variationnelle est représentée par un trait mixte.
Une interprétation possible de cette méthode est de trouver les paramètres du modèle qui donnent la simulation la plus proche des observations réelles. Cela peut aussi s'interpréter comme donner une signification physique aux données observées, les lier dans le temps et dans l'espace par les équations de conservation de la masse, de la quantité de mouvement et de l'énergie.
Le vecteur d'observation $\mathbf{y}$ est souvent de taille plus petite que le vecteur d'état $\mathbf{x}$, d'où la nécessité de prendre en compte les erreurs à l'ébauche pour obtenir un problème bien posé. En pratique, ce terme permet également de régulariser la solution. Nous expliquons son importance plus en détails dans le chapitre 2.

Nous listons, ci-dessous, les avantages et inconvénients des méthodes variationelles :

- ⊕ Si H_{t_i} et M sont linéaires, alors la fonction coût est quadratique et le minimum global existe et il est unique.

⊕ La solution est obtenue après intégration du modèle. Elle est donc cohérente dynamiquement en temps et en espace.
⊕ Cette méthode permet l'estimation des paramètres du modèle ; par exemple, dans ce manuscrit, l'état initial du modèle étant donné la description de la fonctionnelle (1.1).

⊖ Avant toute minimisation, il faut bien définir les matrices de covariances d'erreurs. Elles sont importantes non seulement parce qu'elles influent sur la solution $\mathbf{x}_0^a$ et donc sur l'état analysé $\mathbf{x}^a = M(\mathbf{x}_0^a)$, mais aussi parce qu'elles modifient la nature du système à résoudre. Cependant la connaissance que l'on a sur les covariances d'erreur est en général limitée.
⊖ Les algorithmes de minimisation nécessite la connaissance du gradient de $\mathcal{J}$. Or ce gradient fait apparaître le modèle adjoint (plus de détails sont donnés dans la section 1.3). Dans le cas d'un modèle continu, se pose le problème d'existence de ce modèle adjoint. De plus, si le modèle M est bien posé, il n'en est pas forcément de même pour l'adjoint ; en particulier si l'on tient compte des conditions aux limites réelles des modèles qui sont un problème à elles seules.
⊖ Les méthodes variationnelles ne fournissent pas naturellement d'estimation des erreurs a posteriori $\mathbf{P}^a$. Le calcul est possible mais coûteux (voir le livre de Bennett [2]).
⊖ En situation opérationnelle et en temps réel, on a souvent besoin d'assimiler les observations quand elles arrivent. Les méthodes variationnelles ne sont pas adaptées à l'ajout progressif d'information, il faudra refaire le calcul de minimisation pour garder la cohérence dynamique de la trajectoire optimale. Mais il est possible de réutiliser les résultats des assimilations passées.

1.2.2 Les méthodes séquentielles

Les méthodes séquentielles permettent de mettre à jour facilement la solution assimilée lorsqu'on ajoute une observation. Au niveau opérationnel, elles possèdent donc un grand avantage. Le modèle direct est résolu et à chaque fois qu'une observation arrive (en temps réel), la solution est mise à jour en tenant compte des nouvelles informations, comme illustré sur la figure 1.2 ou sur la figure 1.1 où la méthode est représentée en trait plein.
Cela consiste à remplacer la prévision par une combinaison linéaire de l'état prédit par le modèle

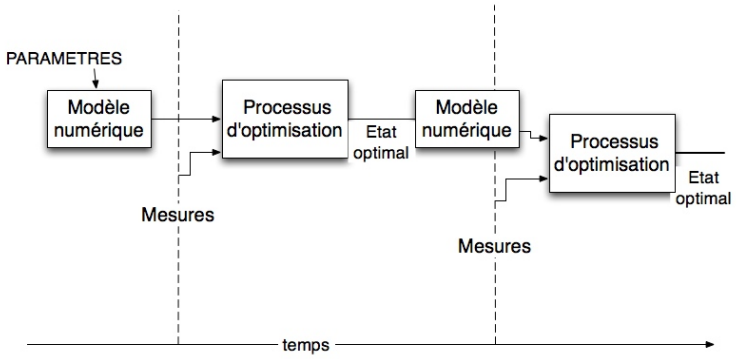

Fig. 1.2 – Principe des méthodes séquentielles : dès qu'une observation est faite, une mise à jour de la solution a lieu.

et le vecteur d'observations. La combinaison linéaire est choisie telle qu'elle minimise la variance de l'erreur de l'état analysé. Ces méthodes sont basées sur les statistiques d'erreurs du modèle et d'erreurs d'observations, d'où leur autre nom "méthodes stochastiques". Introduites en météorologie depuis 1965 [34], elles ont fait l'objet de nombreuses recherches. Citons seulement deux références, une en océanographie [45] et une en météorologie [20].
Pour des modèles linéaires, l'algorithme optimal est le filtre de Kalman (écrit par E.K. Kalman et R. Bucy en 1960 [35]). Il existe également l'interpolation optimale (OI), version simplifiée du filtre de Kalman ([19]), et le nudging ([31], [65]).

Pour des modèles non linéaires, la méthode est adaptée en utilisant des approximations linéaires, mais il faut alors approcher les moments d'ordres élevés de l'équation résolvant la matrice de covariance d'erreur d'analyse $\mathbf{P}^a$. Dans le filtre de Kalman étendu, l'équation donnant la matrice d'erreur de covariance est fermée en supposant que les moments d'ordre supérieurs à 2 sont négligeables. Il s'agit ici non pas de faire une linéarisation dynamique mais une linéarisation statistique.
Cependant, la matrice $\mathbf{P}^a$ est de dimension très élevée et cela complique le calcul et le stockage. Les méthodes de filtre RRSQRT [64] et SEEK [50] sont des variantes du filtre de Kalman étendu qui permettent d'approcher l'espace complet des covariances d'erreur par un sous-espace de dimension réduite en utilisant les modes dominants (filtre de rang réduit). Pour plus de renseignements sur ces variantes, on peut consulter le livre d'Evensen [17].

Les bons résultats pour des modèles simplifiés (quasi géostrophiques, voir Evensen [15]) et le fait que les résultats s'améliorent considérablement si on améliore la prédiction des statistiques d'erreurs ont motivé la recherche de nouvelles méthodes d'estimation de l'évolution de la matrice de covariance d'erreur d'analyse. C'est ce que propose le filtre de Kalman d'ensemble. Présenté par Evensen [16], il consiste à estimer cette matrice en utilisant une méthode de Monte-Carlo. C'est-à-dire que l'on crée un échantillon de plusieurs modèles, ébauches et observations, dont les paramètres auront été perturbés légèrement et à chaque fois différemment. Une prévision est lancée pour chaque jeu (modèle, observation, ébauches). On se retrouve alors avec un échantillon d'état prédit, avec lequel on va estimer la matrice de covariance d'erreur d'analyse.

Nous listons les avantages et inconvénients des méthodes séquentielles :

- ⊕ Dans le cas où le modèle et l'opérateur d'observation sont linéaires, le filtre de Kalman est optimal et est équivalent à l'assimilation variationnelle. En effet, les deux méthodes produisent le même état à la fin de la période de simulation.
- ⊕ Pratiques dans leur implémentation séquentielle, elles ne nécessitent pas le développement du modèle adjoint. Elles permettent, de plus, le calcul des erreurs à posteriori.

- ⊖ Les méthodes séquentielles impliquent le calcul des matrices de covariance d'erreur d'analyse, qui sont de dimensions élevées. En pratique, elles sont estimées à partir d'un échantillon de modèles, faute d'être connues. De plus, les échantillons sont créés à l'aide de modèles simplifiés ou de rang réduit pour réduire le coût de calcul, ce qui ajoute encore à l'imprécision des matrices de covariances.
- ⊖ Les filtres peuvent diverger de fait de la difficile représentation des erreurs.

1.2.3 Les défis à relever

Les méthodes précédentes, séquentielles et variationnelles, sont performantes mais leur utilisation est limitée par le coût de calcul. Elles ont aussi pour défaut de difficilement prendre en compte les non-linéarités des modèles géophysiques. C'est à dire qu'à un moment donné de l'algorithme, faire une hypothèse de linéarité se révèle nécessaire.
Tout aussi difficile à contrecarrer, est leur restriction à des lois statistiques gaussiennes. Jusqu'ici, par simplicité (pour le variationnel) et de par la nature de leur algorithme (pour le séquentiel), l'état optimal est une estimation aux moindres carrés.

Pour relever ces quelques défis, il est possible d'améliorer les méthodes et les algorithmes déjà présentés. Dans cette thèse, par exemple, nous nous intéressons à la réduction du coût de calcul des méthodes variationnelles.
Il est aussi envisageable de développer d'autres méthodes.

Les méthodes purement stochastiques, basées sur des modèles stochastiques, sont de ce point de vue prometteuses.
Nous passons ici à une estimation probabiliste complète de l'évolution du système. Cela va permettre de ne pas faire d'hypothèses sur la quasi-linéarité des équations et de choisir d'autres estimateurs que l'estimation aux moindres carrés. C'est donc une méthode beaucoup plus générale.
La description de cette méthode, est faite dans l'article de Miller *et al* [46]. Elle est ensuite comparée aux méthodes existantes. Le principe général de la méthode est l'utilisation d'un modèle stochastique non-linéaire. On ne résout pas un modèle direct mais on estime la densité de probabilité du système conditionné par les observations.

Dans l'article précédemment cité, des simulations sont faites sur plusieurs problèmes caractéristiques comme le modèle de Lorenz ou un modèle barotropique tronqué spectralement. La conclusion de cet article donne un aperçu du potentiel de cette méthode. En effet, elle peut donner de meilleurs résultats que les filtres de Kalman d'ensemble. Cependant son coût est très élevé et il est difficile de construire les composants aléatoires du modèle stochastique. En effet, on dispose de peu de renseignements sur les bruits que l'on rencontre dans la réalité alors que plus d'informations sont nécessaires que dans les autres méthodes (où on estime seulement les moments).

1.3 Au cœur des méthodes variationnelles

Nous nous intéressons ici plus en détails aux méthodes variationnelles sur lesquelles nous allons appliquer les méthodes multigrilles par la suite. Dans le cas de contraintes fortes, c'est à dire en supposant le modèle parfait, la fonction coût à minimiser est (voir équation (1.1)) :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b]$$

Si le modèle et l'opérateur H sont linéaires, alors $\mathcal{J}$ est une fonction quadratique et définie positive, et on sait qu'il existe une et une seule solution.

Pour résoudre n'importe quel problème d'optimisation avec contraintes, plusieurs possibilités nous sont offertes. Nous pouvons chercher le minimum de la fonction, ou résoudre la formulation duale associée. La formulation duale est basée sur l'inversion des contraintes avec le vecteur de contrôle, et repose sur la résolution du système dit Euler -Lagrange que nous présentons ci après. Nous décrivons ensuite rapidement les méthodes duales puis les méthodes de minimisation dans l'espace primal.

1.3.1 Les équations d'Euler Lagrange

Minimiser $\mathcal{J}$ revient à résoudre $\nabla \mathcal{J}_{\mathbf{x}_0}(\mathbf{x}_0^a) = 0$ et nécessite le calcul du gradient. Cela équivaut à résoudre le système d'Euler-Lagrange associé. Pour l'exprimer, nous allons introduire une variable nommée la variable adjointe. Nous expliquons ici l'apparition de cette variable.

Si on applique une petite perturbation $\delta\mathbf{x}_0$ à $\mathbf{x}_0$, la dérivée de Gâteaux $\hat{\mathcal{J}}(\mathbf{x}_0, \delta\mathbf{x}_0)$ s'écrit alors :

$$\hat{\mathcal{J}}(\mathbf{x}_0, \delta\mathbf{x}_0) =< \nabla_{\mathbf{x}_0}\mathcal{J}, \delta\mathbf{x}_0 > \tag{1.2}$$

En utilisant la définition de la fonction coût, on peut exprimer cette dérivée en fonction de la dérivée de Gâteaux $\hat{\mathbf{x}}$ qui suit l'évolution :

$$\begin{cases} \hat{\mathbf{x}}_{i+1} = \dfrac{\partial M_{i,i+1}}{\partial \mathbf{x}}\hat{\mathbf{x}}_i = \mathbf{M}_{i,i+1}\hat{\mathbf{x}}_i \\ \\ \hat{\mathbf{x}}_0 = \delta\mathbf{x}_0 \end{cases}$$

La dérivée $\hat{\mathcal{J}}$ s'écrit alors :

$$\hat{\mathcal{J}}(\mathbf{x}_0, \delta\mathbf{x}_0) = \sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1} [H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + [\delta\mathbf{x}_0]^T \mathbf{B}^{-1} [\mathbf{x}_0 - \mathbf{x}^b] \tag{1.3}$$

où l'exposant ^{T} signifie qu'il s'agit d'opérateurs adjoints.

L'expression de $\hat{\mathcal{J}}(\mathbf{x}_0, \delta\mathbf{x}_0)$ fait apparaître un terme, difficile à calculer, qui n'est pas fonction directe de $\delta\mathbf{x}_0$.
L'astuce est d'introduire une fonction $\mathbf{p}$ qui va nous permettre de définir $\hat{\mathbf{x}}_i$ en fonction $\mathbf{x}_0$. Pour cela on cherche $\mathbf{p}$ telle qu'elle vérifie

$$\sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T \mathbf{p}_i = \sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T \mathbf{p}_{i+1} + \delta\mathbf{x}_0^T \mathbf{p}_0$$

Or

$$\sum_{i=0}^{N-1} [\hat{\mathbf{x}}_{i+1}]^T \mathbf{p}_{i+1} = \sum_{i=0}^{N-1} [\mathbf{M}_{i,i+1}\hat{\mathbf{x}}_i]^T \mathbf{p}_{i+1} = \sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T . \left[[\mathbf{M}_{i,i+1}]^T \mathbf{p}_{i+1} \right]$$

Donc $\mathbf{p}$ est telle que :

$$\sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T \mathbf{p}_i = \sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T . \left[[\mathbf{M}_{i,i+1}]^T \mathbf{p}_{i+1} \right] + \delta\mathbf{x}_0^T \mathbf{p}_0 \tag{1.4}$$

Nous définissons $\mathbf{p}$ de façon à remplacer le premier terme de droite de l'expression de (1.3) par un terme en fonction de $\delta\mathbf{x}_0$:

$$\begin{cases} \mathbf{p}_i = \mathbf{M}_{i,i+1}^T \mathbf{p}_{i+1} + \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] \\ \mathbf{p}_N = 0 \end{cases}$$

$\mathbf{p}$ est appelée la variable adjointe à $\mathbf{x}$ car elle fait apparaître le modèle adjoint.
On peut alors écrire l'équation (1.4) :

$$\sum_{i=0}^{N-1} [\hat{\mathbf{x}}_i]^T \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] = \delta\mathbf{x}_0^T \mathbf{p}_0$$

Le gradient de $\mathcal{J}$ se trouve en utilisant l'expression ci-dessus dans les définitions (1.3) et (1.2) :

$$\nabla_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) = \mathbf{p}_0 + \mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{1.5}$$

Enfin, l'annulation du gradient nous donne l'expression du minimum $\mathbf{x}_0^a = \mathbf{x}^b + \mathbf{B}\mathbf{p}_0$.

Trouver le minimum de la fonction coût $\mathcal{J}$ revient donc à résoudre le système d'équations suivant, le système d'Euler Lagrange :

$$\begin{cases} \mathbf{x}_{i+1} = M_{i,i+1}(\mathbf{x}_i) + \mathbf{Q}_i \mathbf{p}_{i+1} \\ \mathbf{x}_0 = \mathbf{x}_0^a \\ \\ \mathbf{p}_i = \mathbf{M}_{i,i+1}^T \mathbf{p}_{i+1} - \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] \\ \mathbf{p}_N = 0 \\ \\ \mathbf{x}_0^a = \mathbf{x}^b + \mathbf{B}\mathbf{p}_0 \end{cases} \tag{1.6}$$

La première équation :

$$\mathbf{x}_{i+1} = M_{i,i+1}(\mathbf{x}_i) + \mathbf{Q}_i \mathbf{p}_{i+1} \tag{1.7}$$

constitue ce qu'on appelle le modèle direct, notre modèle d'origine.
Et la dernière équation :

$$\mathbf{p}_i = \mathbf{M}_{i,i+1}^T \mathbf{p}_{i+1} - \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] \tag{1.8}$$

est ce qu'on appelle, le modèle adjoint. L'équation est également appelée modèle rétrograde. En effet, pour résoudre ce système on va du temps $t = t_N$ au temps $t = 0$. On "remonte" dans le temps pour ainsi dire.

Maintenant que le système d'équations à résoudre a été défini, nous remarquons que pour résoudre $\mathbf{x}$, il faut connaître $\mathbf{p}_0$ et pour connaître $\mathbf{p}_0$, il faut résoudre $\mathbf{x}$. Dans le royaume de l'assimilation de données, plusieurs méthodes existent pour résoudre ce système couplé, que l'on regroupe en deux catégories ; les méthodes duales et les méthodes primales.

1.3.2 La formulation duale

Chaque problème d'optimisation admet une formulation duale (Lions, 68 [41]), explicitée par le système d'équations d'Euler Lagrange (1.6). Utiliser la formulation duale permet de minimiser

dans l'espace dual des observations, et de diminuer la taille du vecteur de contrôle, si peu d'observations sont disponibles par rapport à la discrétisation du modèle ; ce qui est souvent le cas. Plusieurs méthodes proposent de résoudre l'assimilation par la formulation duale ; la méthode des représenteurs a été introduite en 1992 par Bennett [2] et par Amodei en 1995 [1] et la méthode nommée 4D-PSAS (Physical-spaceStatistical Analysis System) a été développée par Da Silva *et al.* [12].

Sans entrer plus dans les détails et dans la comparaison de ces différentes méthodes, nous pouvons dire que la formulation duale est équivalente à la formulation primale, si l'opérateur H est linéaire. Nous pouvons faire une liste non exhaustive des avantages et des inconvénients de cette méthode :

- ⊕ Prise en compte des erreurs du modèle sans avoir à augmenter la taille du vecteur de contrôle, contrairement à la résolution dans l'espace primal.
- ⊕ Préconditionnement par $\mathbf{R}^{-1}$ possible pour accélérer la convergence.

- ⊖ Important coût de calcul, dans le cas de la prise en compte des erreurs modèles, même si il y a possibilité de le minimiser.
- ⊖ Difficile à généraliser aux modèles non linéaires ; bien qu'on puisse en trouver un exemple dans [4].

1.3.3 La méthode 4D-variationnelle (4D-Var)

Le 4D-Var résout le problème de l'assimilation de données dans l'espace primal par une méthode itérative dont on peut voir une illustration, figure (1.3). Cette méthode tire son nom de 4D-variationnel, 4D représentant les trois dimensions d'espace et la dimension temporelle. De nombreuses applications à des modèles réalistes ont été effectuées. C'est cette méthode, adaptée aux modèles non linéaires, qui est utilisée dans les systèmes opérationnels de prévision de l'atmosphère de l'European Center for Medium-range Weather Forecasts (ECMWF) depuis 1997 et de Météo France depuis 2000 ; mais également dans les centres de prévisions nationaux aux Etats-Unis (Center for Ocean-Atmosphere Prediction Studies, Florida State University), au Royaume Uni (National Center for Earth Observation, Met Office) au Canada (Environment Canada) et au Japon (Japan Meteorological Agency).
Il s'agit d'estimer itérativement l'état $\mathbf{x}_0^a$ qui minimise $\mathcal{J}$ en utilisant l'information du gradient de $\mathcal{J}$, à l'aide, par exemple, de ce qu'on appelle une méthode de descente :

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} - \alpha_k G_k(\nabla \mathcal{J}(\mathbf{x}_{0,k}))$$

où $\mathbf{x}_{0,k}$, est l'itéré numéro k, $G_k(\nabla \mathcal{J}(\mathbf{x}_{0,k}))$ est la direction de descente et α_k le pas de descente. L'algorithme est appliqué jusqu'à convergence, lorsque $\nabla \mathcal{J}(\mathbf{x}_{0,k}) = 0$.
Pour déterminer G_k et α, on procède de différentes façons selon les algorithmes de descente existants. Les algorithmes sont plus ou moins sophistiqués. Nous parlerons plus en détails de ce point dans la section 2.1.

Dans tous les cas, ces méthodes nécessitent le calcul du gradient. Ce gradient est calculé par la méthode de l'adjoint, proposée par Lions en 1968 [41] et appliquée pour la première fois en météorologie par Le Dimet (1982) [38] et Le Dimet et Talagrand [39].

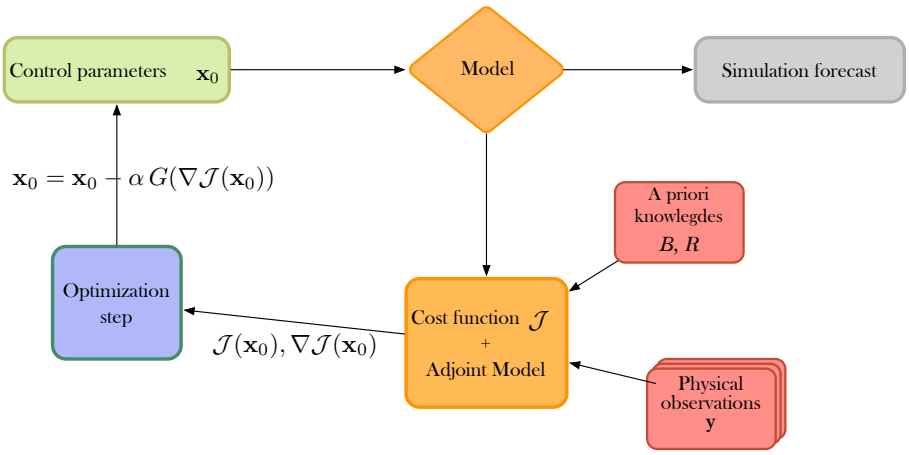

FIG. 1.3 – La résolution itérative de la méthode 4D-Var

Procédons à un petit rappel, on utilise la fonction coût suivante (1.1) :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}]^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T \mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b]$$

Après l'introduction de la variable adjointe $\mathbf{p}$ (voir équation (1.8)),

$$\begin{cases} \mathbf{p}_i = \mathbf{M}_{i,i+1}^T \mathbf{p}_{i+1} - \mathbf{H}_{t_i}^T \mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] \\ \mathbf{p}_N = 0 \end{cases}$$

nous avons pu exprimer le gradient (voir équation (1.5)) :

$$\nabla_{\mathbf{x}_0}\mathcal{J}(\mathbf{x}_0) = \mathbf{p}_0 + \mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b]$$

L'algorithme de résolution du problème 4D-Var est itératif :

1. On pose $\mathbf{x}_{0,k=0} = \mathbf{x}^b$.
2. Résolution du modèle direct (1.7) en partant de $\mathbf{x}_0 = \mathbf{x}_{0,k}$.
3. Résolution du modèle adjoint associé (1.8)
4. Calcul du gradient (1.5)
5. Appel d'une méthode d'optimisation pour trouver $\mathbf{x}_{0,k+1}$.
6. $k + 1 \rightarrow k$
7. Arrêt ou retour à 2.

Comme pour les méthodes duales, le 4D-Var présente des avantages et des inconvénients dont voici une liste non exhaustive :

- ⊕ Préconditionnement possible par $\mathbf{B}^{-1}$.
- ⊕ Sans tenir compte des erreurs du modèle, cette méthode est plus simple et rapide que les méthodes duales.
- ⊕ Adaptation possible aux modèles non linéaires.

⊕ Utilisation importante de cette méthode en opérationnel. De nombreuses recherches sont donc faites, notamment sur la réduction du coût de calcul.
⊕ Utilisation possible d'une fonction coût non quadratique du moment que l'on peut encore définir son gradient. L'estimateur n'est alors plus l'estimateur aux moindre carrés.

⊖ La méthode est très coûteuse si on prend en compte les erreurs du modèle.

1.4 Conclusion

Le monde de l'assimilation de données est vaste mais les travaux présentés ci-après vont porter uniquement sur la méthode variationnelle basée sur le contrôle optimal, et plus particulièrement sur le 4D-variationnel. Cette méthode a l'avantage d'être adaptée à l'estimation de paramètres et d'en donner une formulation dynamique consistante avec la physique. De plus, des applications de ces méthodes ont déjà été effectuées sur des problèmes non-linéaires en océanographie ([18], [44]). Cette méthode est opérationnelle, et quelques modèles d'océans réalistes l'ont implémentée ou vont l'implémenter (ROMS et OPA par exemple). Cependant, l'utilisation de cette méthode demeure difficile et coûteuse, surtout si l'on veut augmenter la résolution des modèles et prendre en compte plus d'observations.
Le prochain chapitre présentera plus en détails les difficultés liées à la résolution de l'assimilation variationnelle ainsi que quelques subterfuges utilisés pour pallier ces difficultés.

2

L'ASSIMILATION VARIATIONNELLE DE DONNÉES : UN SYSTÈME OPTIMAL COMPLEXE

Sommaire

2.1 Résolution du système optimal 40
2.1.1 Le calcul du gradient 40
2.1.2 Les méthodes de minimisation 41
2.2 Les méthodes incrémentales 43
2.2.1 Le principe général 44
2.2.2 La méthode incrémentale vue par le prisme de l'optimisation 46
2.2.3 La méthode incrémentale réduite 47
2.3 Accélérer la convergence 49
2.3.1 Intérêt du préconditionnement 49
2.3.2 Influence de la matrice **B** 51
2.4 Conclusion 52

Nous nous concentrons dans ce chapitre sur la méthode d'assimilation de données variationnelle, le 4D-var, dans le cas d'un modèle supposé parfait ; résolue par méthode de l'adjoint ([38], [39]).
L'assimilation de données variationnelle est basée sur la minimisation du fonctionnelle $\mathcal{J}$ qui mesure l'écart du modèle par rapport aux observations :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{2.1}$$

Le 4D-var consiste à utiliser une méthode itérative pour trouver le minimum de cette fonction coût.

Nous donnons les détails de la résolution du système optimal, et notamment nous présentons, dans le premier paragraphe, les méthodes itératives utilisées. Cela permet de se rendre compte des difficultés liées à la résolution des méthodes variationnelles. Le deuxième paragraphe porte sur l'étude de la méthode incrémentale, et de ses dérivées, qui permettent de simplifier l'optimisation en résolvant une suite de problèmes quadratiques. Cette suite de fonctions à minimiser reste cependant coûteuse en temps de calcul. Le préconditionnement par $\mathbf{B}$, présenté dans le dernier paragraphe, est une méthode permettant d'accélérer la convergence de chacune des minimisations.

2.1 Résolution du système optimal

Nous cherchons le minimum de la fonction coût $\mathcal{J}$, donné par l'équation (2.1) :

$$\mathcal{J}(\mathbf{x}_0^a) = \min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) \tag{2.2}$$

Nous rappelons l'algorithme de la méthode sur la Figure 2.1.

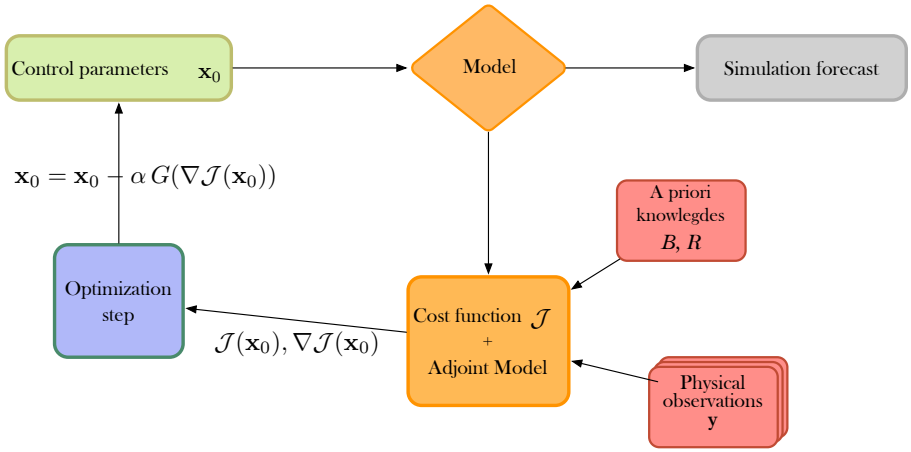

FIG. 2.1 – La résolution itérative de la méthode 4D-Var

Pour trouver le minimum, de nombreux algorithmes "minimiseurs" existent. La plupart de ces algorithmes d'optimisation nécessite de connaître non seulement la valeur de la fonction coût en un "point" mais également la valeur de son gradient $\nabla\mathcal{J}$ en ce même "point".

2.1.1 Le calcul du gradient

Le gradient est calculé par la méthode de l'adjoint déjà présentée section 1.3.3 du chapitre précédent.
Cela implique la discrétisation du modèle adjoint. Ce point peut être délicat si le modèle direct est très complexe. Cela l'est tout particulièrement si les phénomènes simulés comportent des processus à seuil pour lesquels il faut changer de modèles physiques. C'est le cas dans les modèles atmosphériques si l'on passe de l'état vapeur à l'état liquide et lorsqu'il y a une importante

convection dans les modèles de simulation de l'océan, et de l'atmosphère. Les processus à seuil compliquent le calcul de l'adjoint car ils entraînent la perte de la différentiabilité du modèle.
Les modèles peuvent aussi comporter de nombreuses équations à résoudre dont il faut, pour chacune d'elles, écrire le modèle adjoint. Ce qui peut s'avérer laborieux.
Dans notre travail, les modèles utilisés sont suffisamment simples, et les schémas numériques différentiables, pour que nous utilisions un logiciel de différentiation automatique. Ce logiciel s'appelle Tapenade, il est fourni par l'INRIA, et développé par Hascoët et Pascual [27]. Les modèles adjoints donnés par ce logiciel ont pu être vérifiés et testés.

Dans le cas de modèles non linéaires, l'intégration du modèle adjoint nécessite d'avoir en mémoire la solution du modèle direct $\mathbf{x}$ car il en est alors dépendant.
Lorsque la taille des variables et leur nombre sont petits, nous pouvons stocker le résultat du modèle direct en temps et en espace, déjà intégré lors du calcul de la fonction coût. Mais dans le cas de systèmes simulant l'océan ou l'atmosphère, le stockage n'est pas possible étant donné la taille des variables qui atteint très vite un ordre de 10^8. Il s'avère alors nécessaire de mettre en place des techniques (checkpointing) qui nécessitent un recalcul de la solution.
Si on ajoute à cela tous les autres stockages de paramètres nécessaires, le calcul du modèle adjoint et donc du gradient peut avoir un coût sensiblement plus élevé que le modèle direct. Pour des modèles de prévisions géophysiques, ce coût est au pire 7 fois le coût du modèle direct.

2.1.2 Les méthodes de minimisation

La seconde étape délicate de la minimisation est la méthode d'optimisation proprement dite. C'est un algorithme itératif qui permet l'estimation, par une méthode descente, de l'état $\mathbf{x}_0$ qui minimise $\mathcal{J}$ en utilisant l'information du gradient de $\mathcal{J}$:

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} - \alpha_k G_k(\nabla \mathcal{J}(\mathbf{x}_{0,k}))$$

où $\mathbf{x}_{0,k}$ est le $k-$ième itéré, $G_k(\nabla \mathcal{J}(\mathbf{x}_{0,k}))$ est la direction de descente et α_k le pas de descente.

La représentation choisie sur la figure 2.1 est en réalité simplifiée. En pratique, suivant la méthode itérative choisie, le calcul des α_k et des G_k nécessite une ou plusieurs évaluations de la fonction coût et de son gradient. Une itération peut donc demander d'effectuer plusieurs fois la boucle représentée sur la figure : ($\mathbf{x}_0 \Rightarrow$ (Modèle Direct, Modèle Adjoint)$\Rightarrow (\mathcal{J}(\mathbf{x}_0), \nabla \mathcal{J}(\mathbf{x}_0))$).

Le plus simple des algorithmes de minimisation est l'algorithme de la plus grande pente qui estime que la correction à ajouter à $\mathbf{x}_{0,k}$ est proportionnelle à $-\nabla \mathcal{J}(\mathbf{x}_{0,k})$. Dans ce cas G_k est l'identité. La méthode peut utiliser une valeur de pas constant ou calculer un pas optimal à l'aide d'un algorithme de recherche linéaire (plusieurs appels du modèle seront alors nécessaires). Cependant la convergence de l'algorithme de plus grande pente est, en général, peu rapide. Il faut de trop nombreuses itérations pour atteindre le minimum.
D'autres méthodes sont plus efficaces : la méthode du gradient conjugué et les méthodes Quasi-Newton. Nous utiliserons ces deux algorithmes par la suite.

La méthode du gradient conjugué

Cette méthode a été construite pour minimiser les fonctions coût quadratiques du type

$$\mathcal{J} = \frac{1}{2} < \mathbf{x}_0, \mathbf{A}\mathbf{x}_0 > + < \mathbf{b}, \mathbf{x}_0 >$$

où $\mathbf{A}$ est une matrice définie et positive.
De telles fonctions se retrouvent dans le cas d'assimilation de données avec des modèles linéaires.

L'algorithme repose sur la recherche de directions successives $\mathbf{d}_k$ permettant d'atteindre la solution exacte du système étudié. On note $\mathbf{d}_k = G_k(\nabla\mathcal{J}(\mathbf{x}_{0,k}))$. Chaque nouvelle direction doit être $\nabla\mathcal{J}$-orthogonale aux directions précédentes. Pour chaque direction $\mathbf{d}_k$, le pas de descente calculé α_k est tel que l'erreur par rapport à l'état optimal $(\hat{\mathbf{x}}_0 - \mathbf{x}_{0,k+1})$ soit orthogonale à la direction $\mathbf{d}_k$. Chaque itération élimine les composantes en $\mathbf{d}_k$ de l'erreur. Dans le cas de fonctions quadratiques définies positives, l'erreur initiale est composée d'une combinaison linéaire des directions de decente $\mathbf{d}_k$. Si la taille de $\mathbf{x}_0$ est égale à n, le nombre de directions de descente est au plus n. L'algorithme élimine une composante par itération et converge en n itérations.
Voici l'algorithme dans le détail :

1. Pour $k = 0$, $\mathbf{d}_0 = \mathbf{r}_0 = -\nabla\mathcal{J}(\mathbf{x}_{0,k=0})$, le résidu.
2. Calcul du pas de descente
$$\begin{cases} \mathbf{q}_k = \mathbf{A}\mathbf{d}_k \\ \\ \alpha_k = \dfrac{\mathbf{r}_k^T\mathbf{r}_k}{\mathbf{d}_k^T\mathbf{q}_k} \end{cases}$$
3. Calcul du nouvel itéré $\mathbf{x}_{0,k+1}$ et du résidu $\mathbf{r}_{k+1} = -\nabla\mathcal{J}(\mathbf{x}_{0,k+1})$
$$\begin{cases} \mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \alpha_k\mathbf{d}_k \\ \\ \mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k\mathbf{q}_k \end{cases}$$
4. Calcul de la nouvelle direction de descente $\mathbf{d}_{k+1}$:
$$\begin{cases} \beta_{k+1} = \frac{\mathbf{r}_{k+1}^T\mathbf{r}_{k+1}}{\mathbf{r}_k^T\mathbf{r}_k} \\ \\ \mathbf{d}_{k+1} = \mathbf{r}_k + \beta_{k+1}\mathbf{d}_k \end{cases}$$
5. $k = k + 1$ et revenir à (2) jusqu'à convergence

Pour des applications sur des variables de grande taille, la convergence en n itérations n'est pas assurée. Cela est dû au fait que les erreurs de calcul de la machine peuvent devenir importantes et le résidu approcher de moins en moins une direction de descente orthogonale aux précédentes. Enfin, dans le cas de modèles non linéaires, la fonction coût n'est pas quadratique et la méthode doit donc être adaptée. Il existe notamment des variantes avec recherche linéaire.
Face à des problèmes de grande taille et des fonctions coût non quadratiques, il est avantageux d'utiliser d'autres méthodes plus robustes et plus efficaces.

Les méthodes Quasi-Newton

Brièvement, les méthodes de Quasi-Newton englobent toutes les méthodes qui voient G_k comme une approximation de l'inverse de la Hessienne de $\mathcal{J}$, $\mathcal{H} = \nabla^2\mathcal{J}(\mathbf{x}_{0,k})$:

$$G_k = \tilde{\mathcal{H}}_k^{-1}\nabla\mathcal{J}(\mathbf{x}_{0,k}) \simeq \mathcal{H}_k^{-1}(\mathbf{x}_0^k)\nabla\mathcal{J}(\mathbf{x}_{0,k})$$

Le calcul du pas de descente α_k est déterminé le long de la direction de descente, $\mathbf{d}_k = G_k(\nabla\mathcal{J}(\mathbf{x}_{0,k}))$, selon une procédure de recherche linéaire.

L'utilisation de la Hessienne n'est pas anodin. En effet, la méthode est dérivée de la méthode de Newton qui consiste à écrire le développement de Taylor du gradient :

$$\nabla\mathcal{J}(\mathbf{x}_{0,k} + \delta\mathbf{x}_0) = \nabla\mathcal{J}(\mathbf{x}_{0,k}) + \mathcal{H}_k(\mathbf{x}_{0,k})\delta\mathbf{x}_0 + \text{termes aux dérivées supérieures}$$

Pour trouver le minimum de $\mathcal{J}$, on cherche alors $\delta\mathbf{x}_0$ qui vérifie $\nabla\mathcal{J}(\mathbf{x}_{0,k} + \delta\mathbf{x}_0) = 0$. L'équation précédente s'écrit, dans ce cas :

$$\mathcal{H}_k(\mathbf{x}_{0,k})\delta\mathbf{x}_0 = -\nabla\mathcal{J}(\mathbf{x}_{0,k})$$

si on néglige les termes aux dérivées supérieures.

Les méthodes de Quasi-Newton se basent donc sur l'approximation que $\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_0$ vérifie :

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} - \mathcal{H}_k^{-1}(\mathbf{x}_0^k)\nabla\mathcal{J}(\mathbf{x}_{0,k})$$

Nous utilisons la méthode Quasi-Newton développée par l'INRIA, le M1QN3, conçue par Gilbert et Le Maréchal, [21]. L'inverse de la Hessienne est approchée selon une formule L-BFGS (Limited-memory Broyden-Fletcher-Goldfarb-Shanno) de Nocedal, adaptée à une mémoire limitée. L'algorithme intègre également un préconditionneur dont nous étudierons le rôle peu après. Combinée à une recherche linéaire pour le pas de descente, calculée par l'algorithme de Fletcher-Lemaréchal, cette méthode forme la base de la méthode M1QN3.

Ce minimiseur a la particularité d'avoir été conçu pour minimiser des fonctions mettant en jeu des tailles de variables très grandes, comme c'est typiquement le cas pour les problèmes géophysiques.

Les algorithmes présentés ici sont très efficaces dans le cas de modèles linéaires, c'est-à-dire si la fonction coût est quadratique et donc convexe. Ces méthodes permettent également de minimiser des fonctions coût non convexes. Cependant comme ces fonctions peuvent admettre plusieurs points stationnaires, le premier itéré est important et doit être proche de l'optimum. Un algorithme type *simulated annealing* [37], permet de ne pas converger vers un minimum local, sans dépendre de l'information a priori du premier itéré. Des méthodes de région de confiance [24] peuvent être aussi utilisées et sont plus faciles à mettre en œuvre. Cependant, en assimilation de données géophysique, pour des raisons sûrement d'inertie institutionnelle, seuls les algorithmes ci-dessus sont utilisés dans le cadre des méthodes incrémentales que nous présentons dans la section suivante.

2.2 Les méthodes incrémentales

Les équations de la dynamique des fluides sont non linéaires. Les non-linéarités définissent les interactions des échelles entre elles et ne pas les prendre en compte modifie profondément le comportement du système. Les modèles actuels sont donc conçus pour bien représenter les

non-linéarités. De même, les méthodes d'assimilation de données doivent prendre en compte ces non-linéarités. Cependant les fonctions coût à minimiser sont alors non quadratiques et la minimisation devient difficile.
Les méthodes incrémentales permettent de pallier ce problème. En effet, elles reposent sur une méthode de Gauss-Newton, c'est-à-dire une linéarisation globale du modèle et de l'opérateur d'observation autour de l'estimation courante du contrôle $\mathbf{x}_0$. Ceci nous permet d'approcher localement la fonction coût générale (2.1) par une fonction coût quadratique pour trouver l'incrément optimal $\delta\mathbf{x}_0$. Après minimisation, le modèle non linéaire est intégré à partir du nouvel état $\mathbf{x}_0 + \delta\mathbf{x}_0$ et la fonction coût est approchée de nouveau.
L'intérêt de cette méthode est incontestable, puisque pour chacune des fonctions, nous sommes assurés de l'existence et de l'unicité d'un point stationnaire, qui est alors le minimum. Ceci nous permet également d'utiliser des méthodes d'optimisation performantes. Cependant, la convergence dépend de la première estimation et le taux de convergence dépend de validité de l'hypothèse de linéarisation.

Nous nous intéressons ici plus en détails à cet algorithme et à ses propriétés de convergence. Nous présentons également une dérivation de l'algorithme, appelée méthode incrémentale réduite. Cette variante est très utilisée en assimilation de données géophysique car elle permet de réduire et de simplifier le problème initial.

2.2.1 Le principe général

Nous rappelons que l'assimilation de données consiste à minimiser une fonction coût $\mathcal{J}$ avec :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[H_{t_i}(\mathbf{x}_i) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b]$$

Cette fonction est minimisée itérativement par des méthodes d'optimisation usuelles. Cependant, dans le cas où $\mathbf{x}_i$ est calculé suite à l'intégration d'un modèle non linéaire, la fonction $\mathcal{J}$ n'est pas une fonction quadratique.

Pour pallier les problèmes dus aux non-linéarités, Courtier et *al* [11] ont proposé une version incrémentale de la méthode variationnelle. Cette version calcule une suite de minimum $\mathbf{x}_{0,k}$ de fonctions coût quadratiques, en partant de $\mathbf{x}_{0,k=0} = \mathbf{x}^b$.
Pour trouver ces fonctions coût, on linéarise les opérateurs autour de $\mathbf{x}_{0,k}$, et on exprime $\mathcal{J}$ non plus en fonction de $\mathbf{x}_0$ mais en fonction de $\delta\mathbf{x}_0 = \mathbf{x}_{0,k+1} - \mathbf{x}_{0,k}$.
On suppose ainsi que, pour $\mathbf{x}_{0,k+1}$ proche de $\mathbf{x}_{0,k}$, on a :

$$M_{0,i}(\mathbf{x}_{0,k+1}) - M_{0,i}(\mathbf{x}_{0,k}) \simeq \mathbf{M}_{0,i}\delta\mathbf{x}_0$$

$$H_{t_i}(\mathbf{x}_{0,k+1}) - H_{t_i}(\mathbf{x}_{0,k}) \simeq \mathbf{H}_{t_i}\delta\mathbf{x}_0$$

avec $\mathbf{M}_{0,i}$ et $\mathbf{H}_{t_i}$ les linéaires tangents de $M_{0,i}$ et H_{t_i}.
La fonction coût est alors quadratique :

$$\mathcal{J}(\delta\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i]^T\mathbf{R}_i^{-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i] + \frac{1}{2}(\delta\mathbf{x}_0 - \delta\mathbf{x}^b)^T\mathbf{B}^{-1}(\delta\mathbf{x}_0 - \delta\mathbf{x}^b) \quad (2.3)$$

avec

$$\mathbf{d}_i = \mathbf{y}_{t_i} - H_{t_i}(M_{0,i}(\mathbf{x}_{0,k})) \quad \text{et} \quad \delta\mathbf{x}^b = \mathbf{x}^b - \mathbf{x}_{0,k} \tag{2.4}$$

$\mathbf{d}_i$ est appelé vecteur d'innovation ou *misfit*. Il représente l'écart aux observations du modèle non linéaire intégré à partir de l'itéré précédent.
Le nouvel itéré est obtenu par addition de l'incrément optimal $\delta\mathbf{x}_0^a$:

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_0^a$$

Après le calcul du nouvel itéré, on met à jour le *misfit* et l'ébauche $\delta\mathbf{x}^b$, pour construire une nouvelle fonction coût à minimiser. C'est-à-dire que l'on approche de nouveau la fonction coût localement par une autre fonction coût quadratique. Le principe est illustré figure 2.2 et l'algorithme détaillé ci dessous :

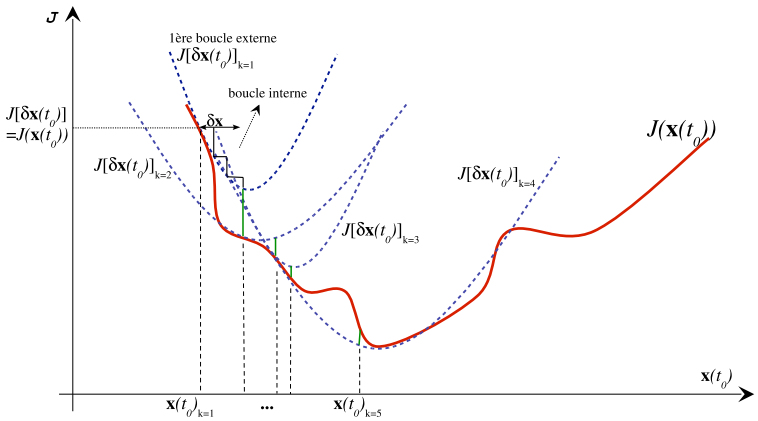

FIG. 2.2 – L'approximation locale par fonction coût quadratique de la méthode incrémentale

Algorithme de la méthode incrémentale :
Données : $\mathbf{x}_{0,k=0} = \mathbf{x}^b$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. $\mathbf{d}_i = \mathbf{y}_{t_i} - H_{t_i}(M_{0,i}(\mathbf{x}_{0,k})), \quad \delta\mathbf{x}^b = \mathbf{x}^b - \mathbf{x}_{0,k}$
2. Chercher le $\delta\mathbf{x}_0^a$ optimal qui minimise $\mathcal{J}(\delta\mathbf{x}_0)$:

$$\mathcal{J}[\delta\mathbf{x}_0] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\delta\mathbf{x}_0 - \delta\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} \tag{2.5}$$

3. Corriger la solution,

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_0^a \tag{2.6}$$

4. $k = k + 1$

Fin de la Boucle externe

Sur la figure 2.3, nous faisons le parallèle avec l'algorithme originel, présenté figure 1.3.
La méthode incrémentale est très utilisée en opérationnel. Des applications ([18], [44]) ont prouvé que son utilisation est valable dans le cas de modèles océan régionaux où les termes non-linéaires ne peuvent être négligés.
Dans la suite, nous présentons ses propriétés de convergence.

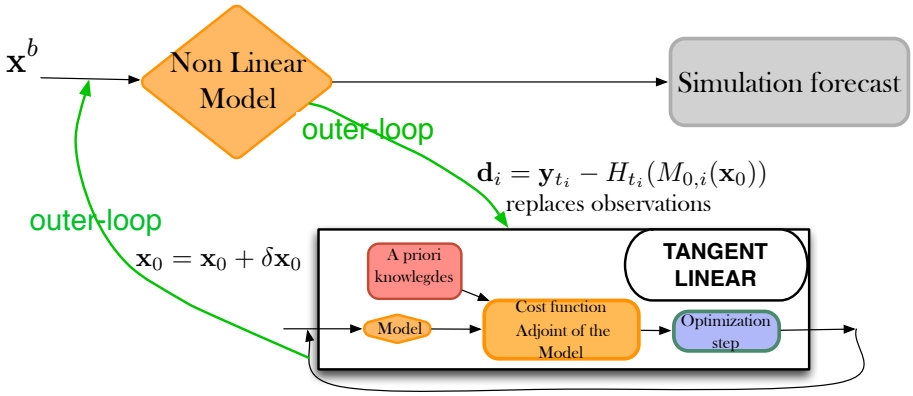

FIG. 2.3 – L'algorithme de la méthode incrémentale

2.2.2 La méthode incrémentale vue par le prisme de l'optimisation

Il est intéressant de regarder la méthode incrémentale d'un point de vue purement optimisation. C'est ainsi qu'on s'aperçoit qu'elle est une application de la méthode Gauss-Newton. Minimiser la fonction coût $\mathcal{J}$ revient à résoudre $\nabla\mathcal{J} = 0$.
La méthode de Newton consiste à faire l'hypothèse de linéarité sur $\nabla\mathcal{J}$ autour de $\mathbf{x}_{0,k}$ et donc à résoudre :

$$\mathcal{H}\delta\mathbf{x}_0 = -\nabla\mathcal{J}(\mathbf{x}_{0,k}) \tag{2.7}$$

au sein de chaque boucle externe, $\mathcal{H}(\mathbf{x}_{0,k}) = \nabla^2_{\mathbf{x}_{0,k}}\mathcal{J}$ étant le Hessien de $\mathcal{J}$.
La méthode de Gauss-Newton part du même principe mais ajoute une hypothèse supplémentaire. Si on écrit la fonction coût sous la forme $\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\|f(\mathbf{x}_0)\|_2^2$ avec f une fonction de $\mathbb{R}^n$ dans $\mathbb{R}^m$, non linéaire et deux fois différentiable.
Minimiser cette fonction $\mathcal{J}$ revient à résoudre $\nabla\mathcal{J} = 0$ ou, dans cet exemple :

$$\mathbf{f}^T f(\mathbf{x}_0) = 0 \tag{2.8}$$

avec $\mathbf{f}(x) = f'(x)$, le Jacobien de f.
La méthode de Newton (2.7) s'écrit alors :

$$\mathcal{H}(\delta\mathbf{x}_0) = \sum_{i=1}^{m} f_i(\delta\mathbf{x}_0).\nabla^2 f_i(\delta\mathbf{x}_0) + \mathbf{f}^T\mathbf{f}(\delta\mathbf{x}_0) = -\nabla\mathcal{J}(\mathbf{x}_{0,k}) = -\mathbf{f}^T f(\mathbf{x}_{0,k}) \tag{2.9}$$

Dans l'équation précédente, les termes d'ordre 2, $\sum_{i=1}^{m} f_i(\delta\mathbf{x}_0).\nabla^2 f_i(\delta\mathbf{x}_0)$, sont coûteux à estimer. L'hypothèse de Gauss-Newton est de les négliger. Le Hessien $\mathcal{H}$ est approché par $\mathbf{f}^T\mathbf{f}$. C'est alors un opérateur linéaire. L'équation résolue devient :

$$\mathbf{f}^T\mathbf{f}\delta\mathbf{x}_0 = -\mathbf{f}^T f(\mathbf{x}_{0,k}) \tag{2.10}$$

Dans le cadre de l'assimilation de données, où l'on doit minimiser la fonction rappelée ci-dessous :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1} \|H_{t_i}(M_{0,i}(\mathbf{x}_0)) - \mathbf{y}_{t_i}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} \tag{2.11}$$

Appliquer la méthode Gauss-Newton revient à résoudre :

$$\left[\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\mathbf{R}_i^{-1}.\,\mathbf{H}_{t_i}\mathbf{M}_{0,i}+\mathbf{B}^{-1}\right]\delta\mathbf{x}_0=-\nabla\mathcal{J}(\mathbf{x}_{0,k}) \tag{2.12}$$

Comme le gradient s'exprime ainsi,

$$\nabla\mathcal{J}(\mathbf{x}_{0,k})=\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\mathbf{R}_i^{-1}.\,\left(H_{t_i}(M_{0,i}\left(\mathbf{x}_{0,k}\right)-\mathbf{y}_{t_i})\right)+\mathbf{B}^{-1}(\mathbf{x}_{0,k}-\mathbf{x}^b),$$

résoudre le système précédent équivaut à minimiser la fonction suivante :

$$\frac{1}{2}\sum_{i=0}^{N-1}\|\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0-(\mathbf{y}_{t_i}-H_{t_i}(M_{0,i}(\mathbf{x}_{0,k})))\|^2_{\mathbf{R}_i^{-1}}+\frac{1}{2}\|\delta\mathbf{x}_0-(\mathbf{x}^b-\mathbf{x}_{0,k})\|^2_{\mathbf{B}^{-1}}$$

On retrouve la fonction coût minimisée par méthode incrémentale, définie en (2.3) :

$$\mathcal{J}[\delta\mathbf{x}_0]=\frac{1}{2}\sum_{i=0}^{N-1}\|\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0-\mathbf{d}_i\|^2_{\mathbf{R}^{-1}}+\frac{1}{2}\|\delta\mathbf{x}_0-\delta\mathbf{x}^b\|^2_{\mathbf{B}^{-1}} \tag{2.13}$$

La méthode incrémentale est donc une méthode de Gauss-Newton.
Il est intéressant de remarquer cette équivalence car, dans le rapport de Gratton *et al.* [23], sont détaillées les propriétés de convergence de la méthode Gauss-Newton.

Théorème 2.2.1. *Soit* $\mathbf{x}_0\in\mathbb{R}^n$. *Si il existe un minimum* $\mathbf{x}_0^a$ *à la fonctionnelle* $\mathcal{J}(\mathbf{x}_0)=\frac{1}{2}\|f(\mathbf{x}_0)\|_2^2$ *et que la matrice* $\mathbf{f}$ *est de rang* n *alors :*

Si $\|1-\mathcal{H}(\mathbf{f}^T\mathbf{f})^{-1}\|<1$, *la méthode Gauss-Newton converge localement vers le minimun* $\mathbf{x}_0^a$.

Selon ce théorème, on remarque que la méthode Gauss-Newton converge vers l'optimum localement, c'est-à-dire si $\mathbf{x}^b$ est suffisamment proche de $\mathbf{x}_0^a$, et si $\mathbf{f}^T\mathbf{f}$ est une bonne approximation du Hessien $\mathcal{H}$.
Un autre théorème affirme que la convergence est au moins linéaire et qu'elle est quadratique si les termes du second ordre du Hessien sont nuls (c'est-à-dire si $\mathcal{H}=\mathbf{f}^T\mathbf{f}$), puisqu'on est dans le cadre d'une méthode de Newton.

2.2.3 La méthode incrémentale réduite

D'un modèle linéarisé à un modèle linéarisé réduit, il n'y a qu'un pas. On franchit ce pas en écrivant $\mathbf{x}_i^s=S(\mathbf{x}_i)=S(M_{0,i}(\mathbf{x}_0))$; avec S un opérateur de simplification et $\mathbf{S}$ son linéaire tangent. Le nouvel algorithme est présenté sur la Figure 2.4 et détaillé ci-dessous. Simplifier le modèle peut avoir plusieurs significations : réduire la résolution et/ou modéliser plus simplement les processus physiques.
L'incrément associé à $\mathbf{x}_i^s$ est

$$\delta\mathbf{x}_i^s=\mathbf{S}\delta\mathbf{x}_i=\mathbf{S}\mathbf{M}_{0,i}\delta\mathbf{x}_0 \tag{2.14}$$

En posant $\mathbf{S}^{-I}$, l'inverse généralisé de S, on peut décrire l'évolution de l'incrément simplifié en fonction de $\delta\mathbf{x}_0^s$:

$$\delta\mathbf{x}_i^s=\mathbf{S}\mathbf{M}_{0,i}\mathbf{S}^{-I}\delta\mathbf{x}_0^s \tag{2.15}$$

Et la nouvelle fonction coût à minimiser dans cet espace simplifié devient :

$$\mathcal{J}[\delta\mathbf{x}_0^s] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}_{t_i}\mathbf{S}^{-I}\delta\mathbf{x}_i^s - \mathbf{d}_i\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\|\delta\mathbf{x}_0^s - (\delta\mathbf{x}^b)^s\|^2_{(\mathbf{SBS}^T)^{-1}} \tag{2.16}$$

avec $(\delta\mathbf{x}^b)^s = S(\mathbf{x}^b - \mathbf{x}_{0,k}) = S(\delta\mathbf{x}^b)$. Le vecteur d'innovation $\mathbf{d}_i$ a toujours la même définition qu'auparavant (2.4). Cette fonction coût remplace la fonction (2.5) dans l'algorithme de la méthode incrémentale.
En pratique les matrices de covariance d'erreur $\mathbf{B}^s = \mathbf{SBS}^T$ et le modèle simplifié $\mathbf{M}^s_{0,i} = \mathbf{SM}_{0,i}\mathbf{S}^{-I}$ sont modélisés directement dans l'espace réduit.

Le résultat de la minimisation est obtenu par addition de l'incrément optimal :

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_0 = \mathbf{x}_{0,k} + \mathbf{S}^{-I}(\delta\mathbf{x}_0^s). \tag{2.17}$$

Cette mise à jour remplace la correction (2.6) de l'algorithme.

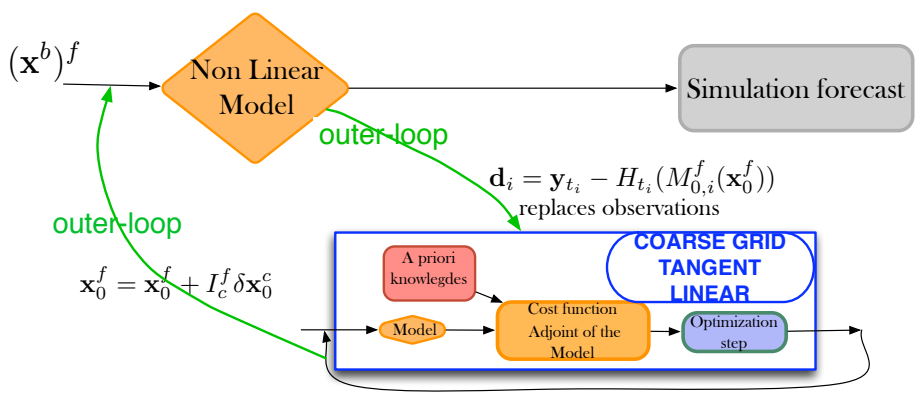

FIG. 2.4 – L'algorithme de la méthode incrémentale réduite

Il s'agit d'une méthode Gauss-Newton "perturbée", c'est-à-dire, que si on écrit $\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\|f(\mathbf{x}_0)\|^2_2$, on ne calcule pas directement $\mathbf{f}$ mais on l'approche par $\widetilde{\mathbf{f}}$. Au lieu de résoudre l'équation (2.10), la méthode de Gauss-Newton perturbée résout :

$$\widetilde{\mathbf{f}}^T\widetilde{\mathbf{f}}\delta\mathbf{x}_0 = -\widetilde{\mathbf{f}}^T f(\mathbf{x}_{0,k}) \tag{2.18}$$

Le système correspondant est alors $\widetilde{\mathbf{f}}^T f(\mathbf{x}_0) = 0$. Gratton *et al.* [23] démontre que la méthode Gauss-Newton "perturbée", présentée ci-dessus, converge sous certaines conditions :

Théorème 2.2.2. *Soit $\mathbf{x}_0 \in \mathbb{R}^n$. Si il existe un minimum $\mathbf{x}_0^{a'}$ tel que $\widetilde{\mathbf{f}}^T f(\mathbf{x}_0^{a'}) = 0$ et que la matrice $\widetilde{\mathbf{f}}$ est de rang n alors :*

Si $\|1 - (\widetilde{\mathbf{f}}^T f + \widetilde{\mathbf{f}}')(\widetilde{\mathbf{f}}^T\widetilde{\mathbf{f}})^{-1}\| < 1$, la méthode converge localement vers le minimun $\mathbf{x}_0^{a'}$.

Selon ce théorème, la méthode incrémentale réduite converge vers une solution $\mathbf{x}_0^{a'}$, différente de $\mathbf{x}_0^a$, localement, c'est-à-dire si $\mathbf{x}^b$ est suffisamment proche de $\mathbf{x}_0^{a'}$. La convergence est au moins linéaire.

Il est possible d'exprimer la distance entre la solution $\mathbf{x}_0^{a'}$ et $\mathbf{x}_0^a$. Elle est bornée par la distance entre les pseudo-inverses de $\widetilde{\mathbf{f}}$ et $\mathbf{f}$.

La méthode incrémentale réduite donne naissance à la méthode multi-incrémentale, utilisée dans les centres de prévisions opérationnelles atmosphériques. La méthode multi-incrémentale utilise, comme modèle réduit, une réduction de dimension et des simplifications physiques. Cela peut s'apparenter aux méthodes multirésolutions que nous étudierons dans ce manuscrit. Elle sera donc décrite plus en détails par la suite. Nous la comparerons aux méthodes multigrilles, qui en sont une alternative.

2.3 Accélérer la convergence

Si nous récapitulons, une itération de minimisation demande une à plusieurs intégrations du modèle direct et du modèle adjoint, qui est au pire 7 fois plus cher que le modèle direct. Chacune des itérations coûte donc plus ou moins 10 fois le modèle direct.
La méthode incrémentale permet de minimiser des fonctions coût quadratiques et donc facilite la convergence. Cependant il faut alors minimiser plusieurs fonctions coût.
En pratique on effectue au plus 3 itérations par boucle externe (voir [61]). Le coût de l'assimilation de données atteint facilement 100 fois le coût du modèle si 3 à 4 mises à jour du modèle non linéaire sont faites.
Accélérer la convergence est donc indispensable pour utiliser ces méthodes au mieux. Une étape importante qui permet de rendre plus efficace la minimisation est le préconditionnement du système. Nous expliquons ici son principe et présentons un préconditionnement couramment utilisé en assimilation de données. Ce préconditionnement fait intervenir la matrice de covariance d'erreur à l'ébauche, $\mathbf{B}$. En plus de son utilisation pour un préconditionnement, cette matrice revêt une importance toute particulière que nous dévoilons ici.

2.3.1 Intérêt du préconditionnement

Si l'on suppose que la fonction à minimiser est quadratique, le principal frein à l'efficacité des méthodes d'optimisation est le conditionnement du Hessien. Le conditionnement du Hessien est défini par le rapport entre la plus grande et la plus petite de ses valeurs propres. Cette quantité est également une mesure de l'ellipticité des iso-surfaces de la fonction coût.
Si ce rapport est grand, la convergence va être plus difficile car l'ellipticité importante donne des fonctions aux courbes peu marquées dont il est plus difficile de trouver le minimum.
Pour une fonction coût quadratique, la solution est donnée par :

$$\mathcal{H}(\delta\mathbf{x}_0) = -\nabla\mathcal{J}(\mathbf{x}_{0,k}) \tag{2.19}$$

avec $\mathcal{H} = \sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\mathbf{R}_i^{-1}\,\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1}$, le Hessien qui est alors une matrice définie positive.
Le conditionnement du problème de minimisation est donc donné par le conditionnement du Hessien.
Il est possible d'accéder aux produit du Hessien-vecteur en utilisant l'adjoint au second ordre [67], qui permet de ne pas expliciter le Hessien. Il existe également des méthodes qui permettent de réduire le conditionnement, sans avoir à calculer les valeurs propres.

Le préconditionnement par la matrice de covariance d'erreur à l'ébauche a été proposé par Courtier [10]. Le changement de variable effectué est alors :

$$\bar{\mathbf{x}}_0 = \mathbf{B}^{-\frac{1}{2}} \delta \mathbf{x}_0$$

Le système résolu (2.19) est équivalent au système préconditionné :

$$\mathbf{B}^{\frac{T}{2}} \mathcal{H} \mathbf{B}^{\frac{1}{2}} (\bar{\mathbf{x}}_0) = -\mathbf{B}^{\frac{T}{2}} \nabla \mathcal{J}(\mathbf{x}_{0,k})$$

En pratique, on a uniquement besoin de connaître $\mathbf{B}^{\frac{1}{2}}$ ou le produit $\mathbf{B}^{\frac{1}{2}}$- vecteur, car la matrice $\mathbf{B}$ est symétrique, pour résoudre

$$\mathbf{B}^{\frac{1}{2}} \mathcal{H} \delta \mathbf{x}_0 = -\mathbf{B}^{\frac{1}{2}} \nabla \mathcal{J}(\mathbf{x}_{0,k})$$

La Hessienne dans l'espace préconditionné devient :

$$\mathcal{H}_{\bar{\mathbf{x}}_0} = \mathbf{B}^{\frac{1}{2}} \mathcal{H} \mathbf{B}^{\frac{1}{2}} = \mathbf{B}^{\frac{1}{2}} \left(\sum_{i=0}^{N-1} [\mathbf{H}_{t_i} \mathbf{M}_{0,i}]^T \mathbf{R}_i^{-1} \mathbf{H}_{t_i} \mathbf{M}_{0,i} \right) \mathbf{B}^{\frac{1}{2}} + \mathbf{I} \tag{2.20}$$

Parce que $\mathcal{J}$ est quadratique, $\mathcal{H}$ est définie positive et $\mathcal{H}_{\bar{\mathbf{x}}_0}$ également. Le préconditionnement nous assure que la plus petite des valeurs propres de $\mathcal{H}_{\bar{\mathbf{x}}_0}$ est supérieure à 1. Le conditionnement est alors borné.
De plus, la taille du vecteur de contrôle étant généralement de taille plus élevée que le nombre d'observations, la Hessienne préconditionnée a une majorité de valeurs propres proches à 1. Le préconditionnement est donc souvent efficace.

Le préconditionnement ainsi défini nécessite le calcul de la racine carrée de $\mathbf{B}$ et peut être difficile à mettre en œuvre. Cela limite le choix des matrices $\mathbf{B}$ et par là-même, les lois statistiques décrivant les erreurs à l'ébauche.
Aussi, le rôle de la matrice $\mathbf{R}$ est également à prendre en compte. Elle représente les statistiques des erreurs de mesures et des erreurs de représentativité.
Les premières sont dues aux instruments de mesures. Les mesures peuvent provenir d'une même plateforme, et/ou de relevés successifs de la même station de mesure. Dans ce cas, les erreurs sont fortement corrélées entre elles pour des relevés proches l'un de l'autre.
Les erreurs de représentativité sont, quant à elles, engendrées par le modèle et la discrétisation. Les erreurs d'interpolation sont corrélées lorsque les observations sont denses par rapport à la résolution du modèle. Enfin les erreurs du modèle, sont propagées par le schéma numérique discret et sont elles aussi corrélées.
Utiliser une matrice $\mathbf{R}$ non diagonale permet de mieux prendre en compte les observations, et leurs différences de valeurs, donc les gradients.
Pourtant, en pratique, la matrice de covariances d'erreurs aux observations $\mathbf{R}$ est une matrice diagonale. Les erreurs sont supposées être non corrélées spatialement et temporellement. Cette hypothèse est faite parce que l'information statistique sur ces erreurs est peu connue ; mais également, parce que prendre une $\mathbf{R}$ non diagonale complexifie la minimisation. En effet, une matrice non diagonale introduit de l'ellipticité plus difficilement réduite par le préconditionnement par $\mathbf{B}$.

Pour faciliter la convergence, le préconditionnement est essentiel. Mais si l'on veut améliorer les représentations des erreurs en allant vers des matrices de covariances des erreurs d'observations plus fidèles, et plus complexes ; le préconditionnement a ses limites d'utilisation et peut ne pas être suffisant.

2.3.2 Influence de la matrice B

Utilisée pour le préconditionnement, la matrice de covariance d'erreur à l'ébauche est un point important de l'assimilation de données.

Dans le cas d'une méthode incrémentale, que ce soit avec préconditionnement ou non, l'incrément optimal $\delta\mathbf{x}_0^a$, calculé après minimisation de la fonction coût (2.5) peut être exprimé selon l'inverse de la Hessienne :

$$\delta\mathbf{x}_0^a = -\left[\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\mathbf{R}_i^{-1}\,\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1}\right]^{-1}\nabla\mathcal{J}(\mathbf{x}_{0,k})$$

avec

$$\nabla\mathcal{J}(\mathbf{x}_{0,k}) = -\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\mathbf{R}_i^{-1}\,\mathbf{d}_i - \mathbf{B}^{-1}\delta\mathbf{x}^b \tag{2.21}$$

Posons $\mathbf{d} = \begin{bmatrix}\mathbf{d}_0\\ \vdots\\ \mathbf{d}_{N-1}\end{bmatrix}$, $\mathbf{M} = \begin{bmatrix}\mathbf{M}_{0,0}\\ \vdots\\ \mathbf{M}_{0,N-1}\end{bmatrix}$, $\mathbf{H} = \begin{bmatrix}\mathbf{H}_0 & (\mathbf{O}) & \dots & (\mathbf{O})\\ (\mathbf{O}) & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & (\mathbf{O})\\ (\mathbf{O}) & \dots & (\mathbf{O}) & \mathbf{H}_{N-1}\end{bmatrix}$ et $\mathbf{R} = \begin{bmatrix}\mathbf{R}_0 & (\mathbf{O}) & \dots & (\mathbf{O})\\ (\mathbf{O}) & \ddots & \ddots & \vdots\\ \vdots & \ddots & \ddots & (\mathbf{O})\\ (\mathbf{O}) & \dots & (\mathbf{O}) & \mathbf{R}_{N-1}\end{bmatrix}$.

La forme (2.21) peut être adaptée en utilisant la formule de Sherman-Morrison-Woddbury [51] :

$$\left[\mathbf{B}^T\mathbf{C}^{-1}\mathbf{B} + \mathbf{A}^{-1}\right]^{-1}\mathbf{B}^T\mathbf{C}^{-1} = \mathbf{A}\mathbf{B}^T\left[\mathbf{B}\mathbf{A}\mathbf{B}^T + \mathbf{C}\right]^{-1}$$

Appliquée au terme contenant $\mathbf{d}$, il vient alors :

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

avec

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

Le terme

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

correspond à une correction dans l'espace des observations calculée comme étant une pondération des "misfit" $\mathbf{d}_i = \mathbf{y}_{t_i} - H_{t_i}(M_{0,i}(\mathbf{x}_0))$ par les covariances d'erreur d'ébauche et d'observation.

La correction sur l'état initial des observations, $\delta\mathbf{x}^a$, peut s'écrire

$$\delta\mathbf{x}^a = \mathbf{B}\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}]^T\hat{\mathbf{d}}_i$$

Ceci signifie que la correction $\hat{\mathbf{d}}_i$ est rapportée sur l'état initial par intégration rétrograde du modèle adjoint. La physique introduite dans le modèle se charge de diffuser temporellement et spatialement l'information corrective aux différentes variables. $\delta\mathbf{x}^a$ est l'image par $\mathbf{B}$ de la somme

de ces différentes corrections. La matrice **B** va donc de nouveau diffuser cette information en espace aux différentes variables de l'état initial.
La définition de la matrice **B** se révèle importante car elle conditionne la correction calculée par l'assimilation de données. Si **B** est exempte de toute loi physique, la correction optimale peut ne pas vérifier certaines lois physiques importantes.
La définition de **B** est donc loin d'être une étape anodine. Nous verrons qu'elle a également un rôle important dans la convergence des algorithmes multigrilles.

2.4 Conclusion

Ce chapitre porte essentiellement sur la mise en évidence des difficultés de résolution de l'assimilation de données et sur son coût prohibitif.

Nous avons présenté le principe des méthodes de minimisation, expliqué pourquoi les méthodes incrémentales peuvent faciliter la convergence, et nous nous sommes intéressés au préconditionnement pour accélérer la convergence.
Malgré toutes ces astuces, l'assimilation de données reste coûteuse. Pour encore réduire le coût, en mode opérationnel, il est utilisé une formulation simplifiée de l'assimilation de données incrémentale, que l'on nomme la méthode incrémentale réduite. Nous avons décrit cette méthode ci-dessus. Lorsque la réduction s'opère sur la taille de discrétisation, cette méthode permet de faire de l'assimilation "haute" résolution, avec un modèle basse résolution. Ce qui a l'énorme avantage de réduire efficacement le coût. En effet si la basse résolution consiste à prendre un point sur trois en chaque dimension d'espace et de temps, la taille des variables est divisée par 3^4 soit 81 à basse résolution.
La méthode incrémentale réduite est théoriquement handicapée car elle ne converge pas vers la solution optimale haute résolution ; de plus les résultats pratiques sont loin d'être satisfaisants [61].
Le prochain chapitre présente les méthodes multigrilles, qui sont des méthodes multirésolutions, que nous allons appliquer à l'assimilation de données ; le but étant d'étudier une alternative à l'assimilation de données sur des modèles multirésolution [14] et aux méthodes incrémentales réduites.

3

LES MÉTHODES MULTIGRILLES

Sommaire

3.1 Les méthodes multigrilles pour la résolution de systèmes linéaires . 54
3.1.1 Les méthodes de relaxation 55
3.1.2 La propriété de lissage 55
3.1.3 Correction par grille grossière 57
3.1.4 Les cycles des méthodes multigrilles 58
3.2 Etude de la convergence 59
3.2.1 L'ellipticité 60
3.2.2 Les opérateurs de transfert 61
3.2.3 Facteurs de convergence 63
3.2.4 La propriété d'approximation 64
3.2.5 Illustration 65
3.2.6 Résumé 67
3.3 Les systèmes non linéaires exposés 67
3.3.1 Les méthodes de résolution classiques 68
3.3.2 La méthode Newton-multigrille (N-MG) 70
3.3.3 L'algorithme Full Approximation Scheme (FAS) 71
3.3.4 Confrontation et hybridation 74

Utilisées pour la résolution de systèmes, les méthodes multigrilles permettent d'améliorer les solutions de méthodes itératives en calculant une correction sur un domaine discrétisé en mailles plus grandes. Ces méthodes sont connues depuis un peu plus de 70 ans (Southwell 1935 [56], 1946 [57]) et n'utilisaient alors seulement que deux niveaux de grilles. Les premiers algorithmes et tests numériques ont été présentés par Brandt, pour la première fois en 1973, et pour une première publication en 1977 [5]. La littérature au sujet des multigrilles est trop conséquente pour autoriser une présentation exhaustive. Historiquement les premiers développements ont été faits sur des systèmes simples, c'est-à-dire linéaires elliptiques de une à deux dimensions, pour

leur simplicité d'analyse. Depuis, le spectre des applications n'a cessé de s'élargir. De nombreux articles étudient les méthodes multigrilles en améliorant les différents éléments qui les constituent ([36], [28], [29], [13], [22]) ou adaptent ces méthodes à différents systèmes, des équations de Poisson, de Stokes ou Navier Stokes compressibles ou incompressibles (où là encore le pionnier est Brandt , 1980 [6], [7]) et différentes conditions aux bords, ou encore au contrôle optimal avec ou sans contraintes (Ta'asan [59]). Le développement des méthodes multigrilles algébriques (Algebraic Multigrid) (McCormick [43]) permet également de considérer des domaines géométriques complexes, et d'adapter plus facilement les multigrilles aux méthodes aux éléments finis. Ceci a ouvert encore un peu plus l'horizon des applications possibles. Le potentiel de ces méthodes est donc élevé et ce chapitre ne présentera qu'une partie de l'iceberg, les fondements des méthodes multigrilles.
Nous avons choisi de commencer la description des méthodes multigrilles, section 3.1, en étudiant une équation de Poisson à une dimension, par souci de simplicité. Puis, dans la section suivante 3.2, nous présentons les critères de choix des différents éléments dont dépend le bon fonctionnement des méthodes multigrilles. Enfin, nous introduisons dans la section 3.3 les algorithmes multigrilles adaptés aux systèmes non linéaires.

3.1 Les méthodes multigrilles pour la résolution de systèmes linéaires

Traditionnellement, les systèmes linéaires sont mis sous forme matricielle, après discrétisation sur un maillage, puis résolus par des méthodes de relaxation. Les méthodes de relaxation sont des méthodes itératives auxquelles on a ajouté un paramètre de relaxation.

Nous présentons ici les méthodes multigrilles, qui sont des algorithmes itératifs basés sur l'utilisation d'une méthode de relaxation dont ils améliorent les propriétés de convergence.
Nous discrétisons un système linéaire sur un maillage régulier à une dimension ; chaque noeud étant espacé d'un autre d'une distance h. On note ce maillage Ω^h. x, la variable d'espace varie en prenant successivement les valeurs $x_j = jh$ avec $0 \leq j \leq N$. N est le nombre de points de la discrétisation.
Le système à résoudre s'écrit alors :

$$\mathbf{A}^h \mathbf{x}_0^{h,a} = \mathbf{g}^h \quad \text{sur } \Omega^h \tag{3.1}$$

$\mathbf{A}^h$ représentant un opérateur discret. Notons $\mathbf{x}_0^h$ l'approximation de la solution exacte $\mathbf{x}_0^{h,a}$. L'erreur algébrique est donnée par $\mathbf{e}^h = \mathbf{x}_0^{h,a} - \mathbf{x}_0^h$, le résidu par $\mathbf{r}^h = \mathbf{g}^h - \mathbf{A}^h \mathbf{x}_0^h$. L'erreur vérifie donc l'équation résiduelle :

$$\mathbf{A}^h \mathbf{e}^h = \mathbf{r}^h \tag{3.2}$$

Résoudre l'équation résiduelle (3.2) revient à résoudre le système originel (3.1).

Nous allons illustrer la méthode multigrille sur l'utilisation d'une méthode de Jacobi pondérée appliquée au système (3.1) en prenant $\mathbf{A}^h$ un opérateur laplacien Δ^h à une dimension, discrétisé en différences finies, à l'aide d'un schéma centré :

$$\Delta^h \mathbf{x}_0^{h,a} = \mathbf{g}^h \quad \text{sur } \Omega^h \tag{3.3}$$

avec $\mathbf{g}^h$ quelconque et des conditions aux bords périodiques.
Ce système permettra l'exposition des principes de fonctionnement des méthodes multigrilles.

3.1.1 Les méthodes de relaxation

Nous considérons la résolution du système (3.1) par une méthode itérative, en partant de $\mathbf{x}^h_{0,k=0}$, première estimation de la solution $\mathbf{x}^{h,a}_0$:

$$\mathbf{M}^h\mathbf{x}^h_{0,k+1} = \mathbf{N}^h\mathbf{x}^h_{0,k} + \mathbf{g}^h, \tag{3.4}$$

avec $\mathbf{A}^h = \mathbf{M}^h - \mathbf{N}^h$.

Par exemple, si on utilise une méthode de Jacobi :

$$\mathbf{N}^h = (\mathbf{L}^h + \mathbf{U}^h), \quad \mathbf{M}^h = \mathbf{D}^h, \tag{3.5}$$

où $\mathbf{D}^h$ est la diagonale de $\mathbf{A}^h$, et $-\mathbf{L}^h$ et $-\mathbf{U}^h$ sont les matrices triangulaires strictement inférieures et supérieures de $\mathbf{A}^h$.
On peut poser $\mathbf{P}^h = (\mathbf{M}^h)^{-1}\mathbf{N}^h = (\mathbf{D}^h)^{-1}(\mathbf{L}^h + \mathbf{U}^h)$ et on a alors $\mathbf{x}^h_{0,k+1} = \mathbf{P}^h\mathbf{x}^h_{0,k} + (\mathbf{M}^h)^{-1}\mathbf{g}^h$.

La méthode Jacobi pondérée de poids $\omega \in [0,1]$, que nous prenons en exemple, est une méthode de relaxation, dont une itération s'écrit :

$$\mathbf{x}^h_{0,k+1} = \left[(1-\omega)\mathbf{I} + \omega\mathbf{P}^h\right]\mathbf{x}^h_{0,k} + \omega(\mathbf{M}^h)^{-1}\mathbf{g}^h. \tag{3.6}$$

avec ω le paramètre de relaxation.
Nous pouvons écrire cette itération autrement, en remarquant que par la définition de $\mathbf{P}^h$, $\mathbf{M}^h$ et $\mathbf{N}^h$:

$$\omega(\mathbf{I} - \mathbf{P}^h) = \omega(\mathbf{I} - (\mathbf{M}^h)^{-1}\mathbf{N}^h) = \omega(\mathbf{M}^h)^{-1}\mathbf{A}^h \tag{3.7}$$

En posant $\mathbf{P}^h_\omega = \left[\mathbf{I} - \omega(\mathbf{M}^h)^{-1}\mathbf{A}^h\right]$, on obtient le processus itératif suivant :

$$\mathbf{x}^h_{0,k+1} = \mathbf{P}^h_\omega\mathbf{x}^h_{0,k} + \omega(\mathbf{M}^h)^{-1}\mathbf{g}^h. \tag{3.8}$$

Dans ce cas, en utilisant la solution exacte vérifiant $\mathbf{A}^h\mathbf{x}^{h,a}_0 = \mathbf{g}^h$, l'erreur $\mathbf{e}^h$ s'écrit :

$$\mathbf{e}^h_{k+1} = \mathbf{x}^{h,a}_0 - \mathbf{x}^h_{0,k+1} = \mathbf{x}^{h,a}_0 - \mathbf{P}^h_\omega\mathbf{x}^h_{0,k} - \omega(\mathbf{M}^h)^{-1}\mathbf{g}^h = (\mathbf{I} - \omega(\mathbf{M}^h)^{-1}\mathbf{A}^h)\mathbf{x}^{h,a}_0 - \mathbf{P}^h_\omega\mathbf{x}^h_{0,k} = \mathbf{P}^h_\omega\mathbf{e}^h_k \tag{3.9}$$

Cette méthode de relaxation converge si et seulement si la plus grande des valeurs absolues des valeurs propres de $\mathbf{P}^h_\omega$ est inférieure à 1. La convergence est d'autant plus efficace que $\omega(\mathbf{M}^h)^{-1}$ est une bonne approximation de l'inverse de $\mathbf{A}^h$. En effet, dans ce cas, $\mathbf{P}^h_\omega$ est proche de la matrice nulle.

3.1.2 La propriété de lissage

Une des caractéristiques de nombreuses méthodes de relaxation est la propriété de *smoothing* ou lissage.
Pour beaucoup de méthodes de relaxation (cf Trottenberg [62]), l'analyse fréquentielle de son facteur de convergence, lorsqu'elle est possible, nous apprend que ces méthodes éliminent efficacement les hautes fréquences de l'erreur. Cependant, une fois ces hautes fréquences éliminées, ces méthodes peinent à converger car elles sont beaucoup moins efficaces à réduire les basses

fréquences restantes. Cette propriété est appelée propriété de *smoothing*.

Pour savoir si une méthode de relaxation vérifie cette propriété, on peut utiliser une analyse de Fourier, c'est-à-dire étudier l'influence d'une itération sur les différentes fréquences de l'erreur. Cela s'effectue en examinant comment réagit la méthode à une perturbation s'exprimant sous la forme d'un mode de Fourier de nombre d'onde m :

$$\mathbf{e}_{k=0}(m, \mathbf{x}^h) = e^{i\,m\mathbf{x}^h}$$

La perturbation $\mathbf{e}_{k=0}(m, \mathbf{x}^h)$ correspond à des grandes échelles si le nombre d'onde m est petit, et à des petites échelles, si m est grand.

Si l'erreur vérifie $\mathbf{e}^h_{k+1} = \mathbf{P}^h_\omega \mathbf{e}^h_k$, le symbole de $\mathbf{P}^h_\omega$, noté $\widetilde{\mathbf{P}_\omega}^h$, permet de connaître la réponse aux perturbations de type $e^{i\,m\mathbf{x}^h}$. En effet, le symbole est la matrice des valeurs propres associées aux modes de Fourier :

$$\widetilde{\mathbf{P}_\omega}^h(m) e^{i\,m\mathbf{x}^h} = \mathbf{P}^h_\omega \left(e^{i\,m\mathbf{x}^h}\right) \quad \forall\, m \tag{3.10}$$

L'analyse de Fourier n'est possible que dans le cas de modèles linéaires et avec des conditions aux bords périodiques ou des domaines infinis.
Nous voulons étudier les propriétés de convergence de la méthode Jacobi pondérée pour résoudre le système (3.3). Dans ce cas $\mathbf{P}^h_\omega$ sur Ω^h est égal à $\left[\mathbf{I} - \omega(\mathbf{M}^h)^{-1}\Delta^h\right]$. $\mathbf{M}^h$ est la diagonale de Δ^h qui est constante. On peut écrire les valeurs propres de $\mathbf{P}^h_\omega$ comme étant égales à $1 - \omega\alpha\lambda(\Delta^h)$ si l'on note α, l'inverse des coefficients de la diagonale de Δ^h et $\lambda(\Delta^h)$ les valeurs propres de Δ^h. On trouve $\alpha\lambda(\Delta^h)(m) = 2\sin^2(\frac{mh}{2})$ et $\widetilde{\mathbf{P}^h_\omega}(m) = |\lambda(\mathbf{P}^h_\omega)(m)| = |1 - 2\omega\sin^2(\frac{mh}{2})|$.

Sur la figure 3.1, on représente le facteur de convergence $\widetilde{\mathbf{P}^h_\omega}$ pour différentes valeurs de ω selon mh, avec un maillage de 100 noeuds intérieurs ($h = 1/100$). Nous rappelons que m petit correspond à des grandes échelles et donc des basses fréquences.

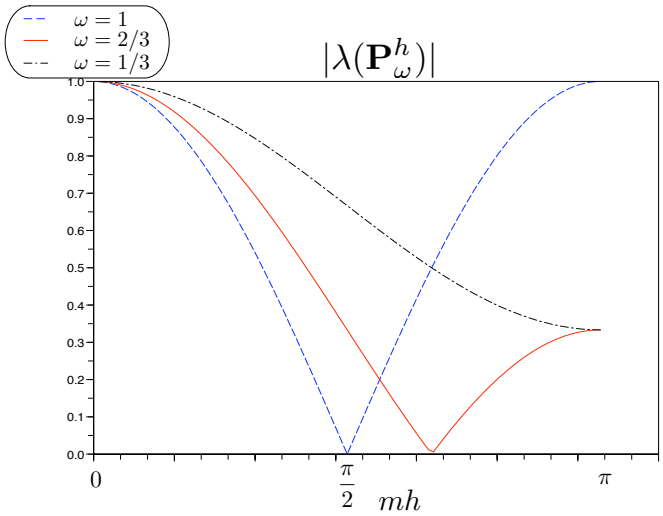

FIG. 3.1 – Le facteur de convergence de la méthode de Jacobi pondérée pour différentes valeurs de ω, en fonction de $mh = [0, \pi[$.

Quelque soit ω, lorsque m est proche de 0, le facteur de convergence est proche de 1. La méthode

converge donc très lentement pour les basses fréquences.
$\omega = \frac{2}{3}$ est le choix de ω qui conduit à la plus forte réduction de l'erreur pour les hautes fréquences ; lorsque $mh > \frac{\pi}{2}$.
La méthode de Jacobi pondérée avec $\omega = \frac{2}{3}$ vérifie donc la propriété de lissage.

3.1.3 Correction par grille grossière

Nous résolvons le système discrétisé (3.1) avec une méthode Jacobi pondérée ($\omega = 2/3$). Cette méthode va peiner à résoudre les basses fréquences, lorsque $mh < \frac{\pi}{2}$. Cependant, ces basses fréquences apparaissent, en partie, comme des hautes fréquences sur une discrétisation moins précise.
C'est de cette idée que vient le principe de fonctionnement des méthodes multigrilles : réduire les hautes fréquences de l'erreur par une méthode de relaxation puis réduire les basses fréquences par ce qui sera désormais nommé "correction par grille grossière".

Explicitons l'idée sur le système linéaire (3.1). La méthode suppose que l'on va travailler avec plusieurs discrétisations. Dans notre exemple, nous considérons uniquement deux résolutions, que l'on note Ω^c et Ω^f. L'exposant c (resp. f) désigne les variables de la grille grossière (resp. fine). Pour échanger les informations entre les deux grilles, on utilise un opérateur de restriction noté I_f^c et un opérateur d'interpolation noté I_c^f. On note $\mathbf{A}^c$ et $\mathbf{A}^f$ les discrétisations de $\mathbf{A}$ et $\mathbf{g}^c$, $\mathbf{g}^f$, celles de $\mathbf{g}$. Sans entrer pour l'instant dans les détails de construction des opérateurs, nous introduisons l'algorithme de la méthode multigrille (voir schéma 3.2) :

Algorithme de la méthode multigrille :
Données : $\mathbf{x}_{0,k=0}^f$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. Relaxation : appliquer ν_1 fois la méthode de relaxation notée SMOOTH à $\mathbf{x}_{0,k}^f$:

$$\mathbf{x}_0^f = SMOOTH^{\nu_1}(\mathbf{x}_{0,k}^f, \mathbf{A}^f, \mathbf{g}^f)$$

2. Correction par grille grossière :
   (a) Résoudre, par une méthode de relaxation ou un solveur direct :

$$\mathbf{A}^c \delta \mathbf{x}_0^c = \mathbf{r}^c, \quad \text{sur } \Omega^c \tag{3.11}$$

   où $\mathbf{r}^c$ est le résidu de la grille fine transféré à basse résolution : $\mathbf{r}^c = I_f^c \mathbf{r}^f = I_f^c(\mathbf{g}^f - \mathbf{A}^f \mathbf{x}_0^f)$.

   (b) Interpoler la correction et en déduire une approximation de la solution sur Ω^f,

$$\mathbf{x}_{0,\text{ after CGC}}^f = \mathbf{x}_0^f + I_c^f \delta \mathbf{x}_0^c \tag{3.12}$$

   after CGC meaning after Coarse Grid Correction.

3. Relaxation : appliquer ν_2 fois la méthode de relaxation pour trouver le nouvel $\mathbf{x}_0^f$:

$$\mathbf{x}_{0,k+1}^f = SMOOTH^{\nu_2}(\mathbf{x}_{0,\text{ after CGC}}^f, \mathbf{A}^f, \mathbf{g}^f)$$

4. $k = k + 1$

Fin de la Boucle

Il est à noter que c'est l'équation résiduelle du système qui est résolue sur la grille basse résolution et non pas le système originel. En effet, la méthode de relaxation sur la grille fine élimine les hautes fréquences de l'erreur et c'est donc l'erreur qu'il faut transposer sur une grille plus grossière pour éliminer les composantes basses fréquences restantes.
La troisième étape, qui consiste à effectuer quelques itérations de minimisation sur la grille fine, est nécessaire car le processus itératif n'a pas convergé. La correction par grille grossière élimine les basses fréquences de l'erreur mais il reste encore des hautes fréquences que les ν_1 premières itérations n'ont pas forcément suffisamment réduites. De plus, l'interpolation, qui n'est pas exacte, de l'erreur à basse résolution fait apparaître des hautes fréquences sur la grille fine.

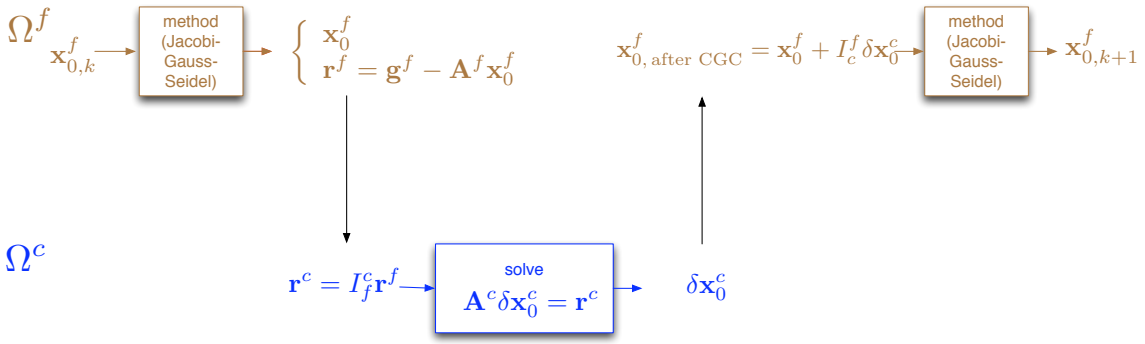

FIG. 3.2 – L'algorithme des méthodes multigrilles

3.1.4 Les cycles des méthodes multigrilles

Nous venons de décrire le principe des méthodes multigrilles avec une version de deux grilles. Cependant, en pratique, il est utilisé plus de deux résolutions.
L'extension se fait de manière récursive, puisqu'on peut, pour résoudre l'équation résiduelle, utiliser une méthode multigrille. Cela permet d'une part de diminuer le coût de calcul si la méthode directe est très coûteuse. D'autre part, augmenter le nombre de résolutions permet aussi de diminuer la taille du domaine fréquentiel correspondant aux hautes fréquences pour chaque résolution. Ce qui rend plus rapide la convergence.
Le nombre de possibilités d'algorithme augmente avec le nombre de grilles.
Pour une utilisation $2-$grilles, on se restreint au V-cycle décrit précédemment. Quand on ajoute une grille, le choix est beaucoup plus vaste. Un choix commun est d'utiliser le même nombre γ d'appels de corrections par grille grossière pour chaque résolution d'équations résiduelles.

Ces exemples (illustration 3.3) et bien d'autres peuvent être trouvés dans le livre de W. Briggs [9] ou de Brandt [8] ou dans celui de Trottenberg *et al* [62], plus complet et complexe.
À tous ces algorithmes, il est possible d'ajouter l'option d'un premier passage sur les grilles grossières pour calculer une première estimation de la solution de la grille fine. Là encore plusieurs choix sont possibles. On peut résoudre le système de la plus basse à la plus haute résolution ou

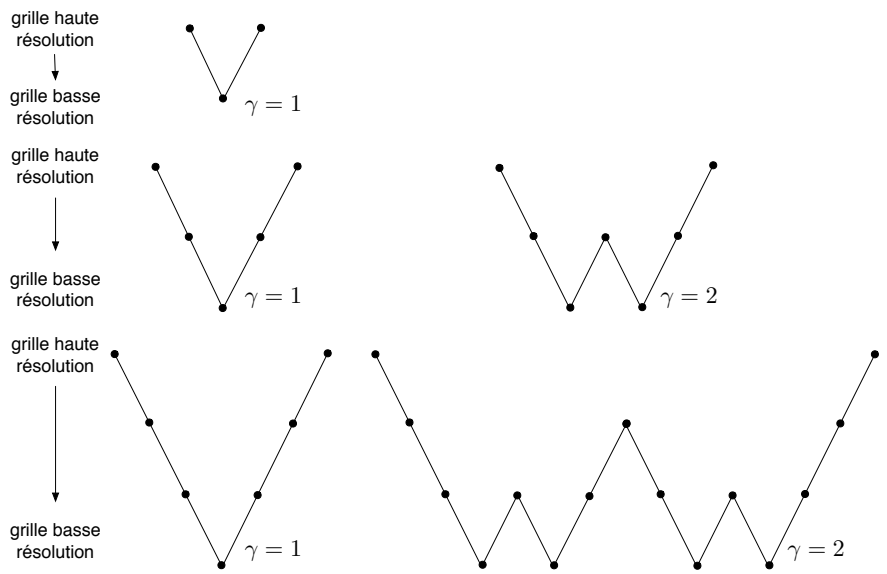

FIG. 3.3 – Exemples de parcours des différentes grilles dans le cas de 2, 3 ou 4 grilles.

utiliser l'algorithme couramment nommé FullMultigrid (ou FMG). Le FMG consiste à résoudre quelques itérations du système sur la grille la moins précise avant de transférer la solution sur chaque grille de proche en proche jusqu'à la plus fine. Entre chaque transfert, on effectue un cycle d'un algorithme multigrille pour résoudre le système (voir illustration 3.4).

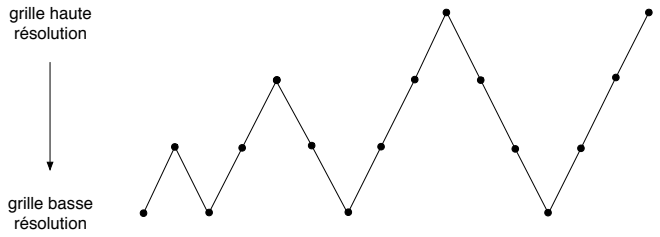

FIG. 3.4 – L'algorithme FMG ; dans le cas de 4 grilles.

3.2 Etude de la convergence

Dans le rapport technique de Hemker [30], dans le manuel sur les méthodes multigrilles de Trottenberg [62], ou encore dans le livre de Hackbusch [26], il est expliqué qu'en utilisant l'analyse de Fourier, on peut étudier le comportement fréquentiel des méthodes multigrilles dans leur ensemble, à condition de connaître ou de pouvoir calculer le symbole de la méthode de relaxation et des opérateurs de transfert.
Ce type d'étude a permis de mettre en évidence certains critères de choix pour les méthodes de relaxation, les opérateurs de transfert et le système grossier $\mathbf{A}^c$, que nous allons présenter maintenant.

3.2.1 L'ellipticité

Nous commençons ce paragraphe en introduisant un concept important dans le cadre des méthodes multigrilles, la h-ellipticité. Il s'agit de mesurer l'ellipticité du système à résoudre, sous sa forme discrète. Cette notion est une condition nécessaire mais non suffisante à l'efficacité de la méthode de relaxation. On ne s'intéresse ici qu'à une grille.

Qu'entend-t'on par h-ellipticité ?
Définissons $m_{\text{high}} = \left[-\frac{\pi}{h}, \frac{\pi}{h}\right[\setminus \left[-\frac{\pi}{2h}, \frac{\pi}{2h}\right[$ comme étant les composantes hautes fréquences représentées par la grille Ω^h, régulière dont les mailles sont de taille h. On rappelle que $\widetilde{\mathbf{A}}^h$ est le symbole de l'opérateur discrétisé, $\mathbf{A}^h$ (voir définition (3.10)).
La h-ellipticité est calculée à l'aide de la mesure $E_h(\mathbf{A}^h)$, définie de cette façon :

$$E_h(\mathbf{A}^h) = \frac{\min\left\{|\widetilde{\mathbf{A}^h}(m)|, m \in m_{\text{high}}\right\}}{\max\left\{|\widetilde{\mathbf{A}^h}(m)|, m \in]-\frac{\pi}{h}, \frac{\pi}{h}[\right\}} \tag{3.13}$$

Par définition, $\boxed{\mathbf{A}^h \text{ est } h\text{-elliptique si } E_h(\mathbf{A}^h) > 0.}$

Dans le cas où $\mathbf{A}^h$ n'est pas h-elliptique, $E_h(\mathbf{A}^h) = 0$ et il existe un mode $m \in m_{\text{high}}$ pour lequel le symbole $\widetilde{\mathbf{A}}^h(m)$ est nul. On ne peut alors pas trouver de méthode itérative qui présente une convergence plus rapide pour les hautes fréquences que pour les basses.
En effet, supposons qu'on utilise une méthode de relaxation, type Jacobi pondéré qui nous donne cette valeur de la prochaine erreur :

$$\mathbf{e}_{k+1}^h = \mathbf{P}_\omega^h \mathbf{e}_k^h$$

avec $\mathbf{P}_\omega^h = \mathbf{I} - \omega(M^h)^{-1}\mathbf{A}^h$.
Donc si $\mathbf{A}^h$ n'est pas h-elliptique, il existe un $m \in m_{\text{high}}$ tel que $\widetilde{\mathbf{A}}^h(m) = 0$ et tel que $\widetilde{\mathbf{P}}_\omega^h(m) = 1$.
Ce qui implique que le facteur de convergence est au moins égal à 1 pour une des hautes fréquences. Pour cette fréquence, la méthode de relaxation ne peut pas converger.
La h-ellipticité est une condition nécessaire mais non suffisante quant à la convergence de la méthode.

Conclusion :
Il faut donc d'abord étudier l'ellipticité du système à résoudre pour être sûr que l'on puisse trouver une méthode de relaxation, qui a la propriété de lissage.

Prenons maintenant l'exemple du système (3.3), on sait $|\tilde{\Delta}^h(m)| = \frac{4}{h^2}\sin^2(\frac{mh}{2})$.
L'opérateur est bien elliptique, c'est pourquoi on peut trouver un facteur de convergence inférieur à 1 pour les hautes fréquences, avec , par exemple $\omega = 2/3$ (voir figure 3.1).

La propriété de h-ellipticité peut se traduire d'une autre manière (voir Ta'asan [59]). Un opérateur h-elliptique réagit à une composante haute fréquence par un changement à haute fréquence local.
Dans un cadre général : une fonction $F : \mathbb{R}^N \Rightarrow \mathbb{R}$ est dite elliptique, si elle est deux fois dérivable et sa Hessienne est minorée au sens des formes quadratiques, par un multiple de l'identité :

$$\exists c > 0, \forall x \in \mathbb{R}^N, \quad \forall u \in \mathbb{R}^N, \quad < \mathbf{x}, \nabla^2 F(x)\mathbf{x} > \geq c\|\mathbf{x}\|^2 \tag{3.14}$$

Toute constante c vérifiant l'inégalité précédente est alors appelée constante d'ellipticité de F.

L'ellipticité d'un opérateur peut également se définir sous la forme fréquentielle, en utilisant le symbole. Ainsi, dans le cas continu, $\mathbf{A}$ est elliptique si :

$$|\tilde{\mathbf{A}}(m)| \geq C \sum_p |m|^{2p}, \quad p > 0 \tag{3.15}$$

Et dans le cas discret, $\mathbf{A}^h$ est h-elliptique si :

$$|\tilde{\mathbf{A}}^h(m)| \geq C \sum_p \sin\left(\frac{mh}{2}\right)^{2p}, \quad p > 0 \tag{3.16}$$

avec C une constante et m, dans le cas discret, variant entre $[-\frac{\pi}{h}, \frac{\pi}{h}]$.
Ce sont ces deux dernières définitions que nous utiliserons par la suite.

3.2.2 Les opérateurs de transfert

Le rapport technique de Hemker [30] nous indique que le choix de l'opérateur de transfert est conditionné par le système à résoudre. Avant d'expliciter ces critères de choix, nous présentons les plus simples opérateurs d'interpolation.
Nous nous plaçons dans le cas de deux maillages tels décrits dans la première section Ω^f et Ω^c, supposés à une dimension. Nous supposons également que le raffinement est d'ordre 2 entre ces deux maillages.

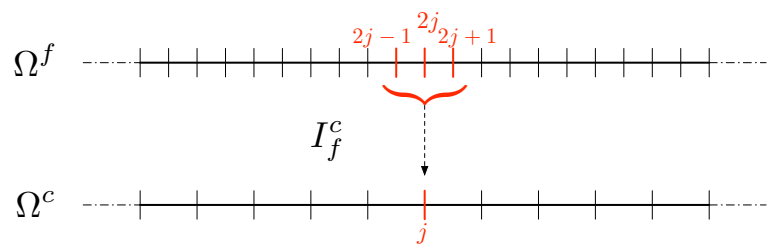

FIG. 3.5 – Transfert de la grille fine vers la grille grossière, par l'opérateur "full-weighting", une dimension

Pour transposer l'erreur de la grille fine à la grille grossière, on peut, par exemple, utiliser l'opérateur de full-weighting (voir Figure 3.5) :

$$\mathbf{x}_0^c(j) = \frac{1}{4}\mathbf{x}_0^f(2j-1) + \frac{1}{2}\mathbf{x}_0^f(2j) + \frac{1}{4}\mathbf{x}_0^f(2j+1), \quad \forall j \tag{3.17}$$

Dans ce cas, lorque l'erreur est régulière, sa restriction est régulière également, mais lorsque l'erreur oscille trop fortement, la restriction ne donne pas une bonne représentation de l'erreur ; ce que l'on appelle le phénomène d'aliasing.

Pour interpoler la mise à jour de Ω^c à Ω^f, on utilise un opérateur d'interpolation I_c^f. Le plus simple est l'interpolation linéaire (voir Figure 3.6) :

$$\begin{cases} \mathbf{x}_0^f(2j) = \mathbf{x}_0^c(j) \\ \mathbf{x}_0^f(2j+1) = \frac{1}{2}(\mathbf{x}_0^c(j) + \mathbf{x}_0^c(j+1)) \end{cases}, \quad \forall j \tag{3.18}$$

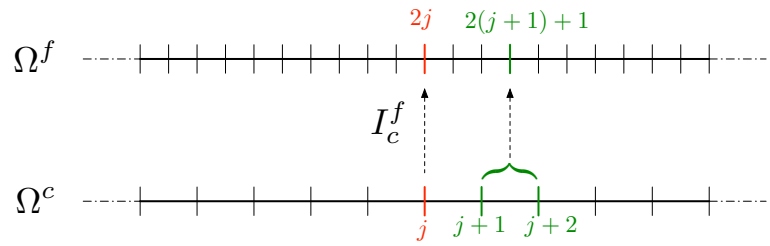

FIG. 3.6 – Transfert de la grille grossière vers la grille fine, par l'opérateur d'interpolation linéaire, une dimension

Là aussi, si la fonction sur la grille Ω^c est régulière, elle apparaîtra régulière sur la grille Ω^f mais des hautes fréquences sont introduites due aux défauts de l'interpolation.
Les opérateurs de transfert peuvent être différenciés par deux de leurs caractéristiques : l'ordre premier (ou l'ordre basse-fréquence) d'un opérateur de transfert et l'ordre second (ou l'ordre haute-fréquence). Nous allons définir ci-après, ces notions et leur rôle dans le choix des opérateurs de transfert.

L'ordre premier :
L'ordre premier ou ordre basse-fréquence d'un opérateur d'interpolation (resp. de restriction), que l'on note I_c^f (resp. I_f^c) est le plus grand nombre $\alpha \geq 0$ tel que :

$$\widetilde{I_c^f}\,(m) = 1 + \mathcal{O}(|m|^\alpha) \quad |m| \to 0$$

L'ordre second :
Pour un facteur de raffinement $q \in \mathbb{Z}$, l'ordre second ou ordre haute-fréquence d'un opérateur d'interpolation (resp. de restriction), que l'on notera de manière générale I_c^f (resp. I_f^c) est le plus grand nombre $\alpha \geq 0$ tel que :

$$\widetilde{I_c^f}\left(m + \frac{2\pi p}{qh}\right) = \mathcal{O}(|m|^\alpha) \quad |m| \to 0$$

pour chaque $p \in]0, q] \subset \mathbb{Z}$.

L'ordre α de l'interpolation (resp. la restriction) nous indique que les polynômes d'ordre $\alpha - 1$ seront interpolés exactement par I_c^f (resp. par $(I_f^c)^T$).

On peut calculer le symbole de l'interpolation linéaire, avec un raffinement égal à 2 et sur une grille à une dimension de taille h :

$$\widetilde{I_f^c}(m) = \frac{1}{4}\left(e^{-imh} + 2 + e^{imh}\right)$$

ce qui est égal à

$$\widetilde{I_f^c}(m) = \frac{1}{4}\left(2 + 2\cos(mh)\right) = 1 - \sin^2(\frac{mh}{2}) \tag{3.19}$$

L'ordre premier, déductible à partir de l'expression (3.19), est égal à 2. L'ordre second se calcule à partir de

$$\widetilde{I_f^c}(m + \frac{\pi}{h}) = \frac{1}{2}\left(1 + \cos(mh + \pi)\right) = \frac{1}{2}\left(1 - \cos(mh)\right)$$

L'ordre second est également égal à 2.

Les opérateurs de transfert sont à choisir selon l'ordre du système d'équations différentielles à résoudre.
Supposons qu'il s'agisse d'un opérateur d'ordre M. Dans [30] (Hemker), il est prouvé que, si on note m_1, et m_2 (resp. n_1, n_2) ; les premier et second ordres de l'interpolation (resp. la restriction), alors pour avoir convergence d'un passage par grille grossière, il faut vérifier :

$$m_1 > 0, n_1 > 0; \quad \text{et} \quad n_2 + m_2 \geq M \tag{3.20}$$

Dans notre exemple en laplacien (3.3), nous avons affaire à un opérateur d'ordre 2.
Or l'ordre second d'un opérateur d'interpolation linéaire, ainsi que celui de l'opérateur de restriction "full-weighting", sont égaux à 2 ($n_2 = m_2 = 2$) car le full-weighting est le transposé de l'interpolation linéaire à un coefficient multiplicatif près. On peut donc utiliser une interpolation linéaire et un full-weighting pour le cas laplacien-Jacobi pondéré, sans dégrader les résultats de convergence ($n_2 + m_2 \geq M = 2$). On peut également se contenter d'un opérateur de restriction comme l'injection, pour lequel $m_2 = 0$.

Nous nous limitons dans ce manuscrit à des opérateurs de transfert simples. Il existe cependant de nombreux opérateurs d'ordres plus élevés, ou possédant des propriétés de conservation intéressantes, qui peuvent être par exemple des filtres d'ondelettes (Goedecker *et al.* [22]).

3.2.3 Facteurs de convergence

L'efficacité du passage par la grille grossière en lui-même est donnée par l'étude du facteur de convergence d'une correction par grille grossière.
D'après l'algorithme des méthodes multigrilles, section 3.1.3, on peut exprimer le passage par la grille grossière (voir équation (3.12)) :

$$\mathbf{x}^f_{0,\,\text{after CGC}} = \mathbf{x}^f_0 + I^f_c \delta \mathbf{x}^c_0$$

Or $\delta \mathbf{x}^c_0$ est solution de l'équation résiduelle (3.11) : $\mathbf{A}^c \delta \mathbf{x}^c_0 = -I^c_f \mathbf{A}^f \mathbf{x}^f_0 + I^c_f \mathbf{g}^f$.

Si on suppose que $\mathbf{A}^c$ est inversible, on note $(\mathbf{A}^c)^{-1}$ son inverse, alors on peut simplifier l'expression de l'erreur en :

$$\delta \mathbf{x}^c_0 = -(\mathbf{A}^c)^{-1} I^c_f \mathbf{A}^f \mathbf{x}^f_0 + (\mathbf{A}^c)^{-1} I^c_f \mathbf{g}^f \tag{3.21}$$

On obtient l'effet sur l'état de la grille fine d'un passage par la grille grossière :

$$\mathbf{x}^f_{0,\,\text{after CGC}} = \left(\mathbf{I} - I^f_c (\mathbf{A}^c)^{-1} I^c_f \mathbf{A}^f\right) \mathbf{x}^f_0 + I^f_c (\mathbf{A}^c)^{-1} I^c_f \mathbf{g}^f \tag{3.22}$$

Comme l'erreur s'exprime selon l'égalité $\mathbf{e}^f = \mathbf{x}^{f,a}_0 - \mathbf{x}^f_0$, on peut obtenir son évolution après un passage par la grille grossière :

$$\mathbf{e}^f_{\text{after CGC}} = \mathbf{x}^{f,a}_0 - \mathbf{x}^f_{0,\,\text{after CGC}} = \mathbf{x}^{f,a}_0 - \left(\mathbf{I} - I^f_c (\mathbf{A}^c)^{-1} I^c_f \mathbf{A}^f\right) \mathbf{x}^f_0 - I^f_c (\mathbf{A}^c)^{-1} I^c_f \mathbf{g}^f$$

Or $\mathbf{g}^f = \mathbf{A}^f \, \mathbf{x}^{f,a}_0$, l'erreur vérifie plus simplement :

$$\mathbf{e}^f_{\text{after CGC}} = \left(\mathbf{I} - I^f_c (\mathbf{A}^c)^{-1} I^c_f \mathbf{A}^f\right) \mathbf{e}^f \tag{3.23}$$

L'opérateur itératif associé à une correction par grille grossière, que l'on note $\mathbf{K}$, s'écrit donc :

$$\mathbf{K} = \mathbf{I} - I_c^f (\mathbf{A}^c)^{-1} I_f^c \mathbf{A}^f \tag{3.24}$$

Une correction par grille grossière est donc équivalente à une méthode de relaxation qui approche $(\mathbf{A}^f)^{-1}$ par $(I_c^f (\mathbf{A}^c)^{-1} I_f^c)^{-1}$. On peut alors étudier la convergence d'un passage par la grille grossière en étudiant les valeurs propres de $\mathbf{K}$.

Nous pouvons également obtenir le facteur de convergence global d'un algorithme V-cycle des méthodes multigrilles, à l'aide du facteur de convergence de la méthode de relaxation $\mathbf{P}_\omega$, défini section 3.1.1 :

$$\mathbf{K}_{\text{global}} = (\mathbf{P}_\omega^f)^{\nu_2} \mathbf{K} (\mathbf{P}_\omega^f)^{\nu_1} \tag{3.25}$$

ν_1 est le nombre d'itérations de la méthode de relaxation sur la grille fine avant le passage par la grille grossière et ν_2 le nombre d'itérations de cette même méthode après le passage par la grille grossière.

Hackbusch [26] analyse ce facteur global en se concentrant sur le cas où $\nu_2 = 0$. Il factorise alors $\mathbf{K}_{\text{global}}$:

$$\mathbf{K}_{\text{global}} = \left[(\mathbf{A}^f)^{-1} - I_c^f (\mathbf{A}^c)^{-1} I_f^c \right] \left[\mathbf{A}^f (\mathbf{P}_\omega^f)^{\nu_1} \right]$$

Cette factorisation lui permet de déduire des conditions sur le passage par la grille grossière selon le comportement de la méthode de relaxation. C'est ce qu'il appelle la propriété d'approximation que nous explicitons dans la sous section suivante.

3.2.4 La propriété d'approximation

Supposons que la méthode de relaxation possède la propriété de lissage, déjà explicitée section 3.1.2. Selon Hackbusch [26], elle vérifie alors :

$$\exists\, \alpha, \quad \|\mathbf{A}^f (\mathbf{P}_\omega^f)^\nu\| \leq \eta(\nu)(\Delta x^f)^{-\alpha}, \forall \nu \geq 1, \quad \text{avec} \lim_{\nu \to \infty} \eta(\nu) = 0 \tag{3.26}$$

Le coefficient α dépend de la méthode itérative choisie et du système résolu. Il est, par exemple, égal à l'ordre du système $\mathbf{A}$, dans le cas d'une méthode Jacobi pondérée avec l'utilisation de normes euclidiennes.

L'opérateur de la grille grossière $\mathbf{A}^c$ et les opérateurs de transfert doivent, quant à eux, vérifier la propriété d'approximation (Hackbusch [26]) :

Théorème 3.2.1. *On résout le système $\mathbf{A}^f \mathbf{x}_0^{f,a} = \mathbf{g}^f$ sur la grille Ω^f. Soient I_f^c et I_c^f les opérateurs de restriction et d'interpolation entre la grille Ω^f et la grille Ω^c à plus basse résolution et Δx^f le pas d'espace de la grille fine.*
Alors, si la méthode de relaxation $\mathbf{P}_\omega^f$ vérifie la propriété de lissage (3.26) et s'il existe $C > 0$ tel que

$$\|(\mathbf{A}^f)^{-1} - I_c^f (\mathbf{A}^c)^{-1} I_f^c\| < C(\Delta x^f)^{\alpha'}$$

avec $\alpha' \geq \alpha$, alors il y a convergence de la méthode multigrille.

Il est important de repréciser que ce résultat est valable uniquement si l'équation résiduelle

de la grille grossière est résolue de manière exacte.

Le respect de la propriété d'approximation nous donne des conditions sur les ordres des opérateurs de restriction et d'interpolation et impose aux opérateurs $\mathbf{A}^f$et $\mathbf{A}^c$ d'être des discrétisations consistantes de l'opérateur $\mathbf{A}$.
Ainsi, l'opérateur $\mathbf{A}^c$ doit être construit de telle manière qu'il vérifie l'approximation suivante :

$$\|\mathbf{A}^c - I_f^c \mathbf{A}^f I_c^f\| \leq (\Delta x^f)^{\alpha'} \tag{3.27}$$

Le coefficient α' est supérieur ou égal à p, l'ordre de consistance des discrétisations. Généralement nous trouvons donc $\alpha^{'} \geq \alpha$ et il y a convergence de la méthode multigrille.

La première méthode de construction de $\mathbf{A}^c$ qui vient à l'esprit est de prendre le même opérateur que $\mathbf{A}^f$ mais en modifiant la valeur de la taille de discrétisation. Ceci est particulièrement facile pour les systèmes d'équations aux dérivées partielles discrétisées en différences finies. Mais d'autres méthodes sont possibles.
Si l'on a affaire à la reproduction de phénomènes physiques, on peut ne pas faire les mêmes approximations selon la taille de la discrétisation. Par exemple, la viscosité peut être négligée à très haute résolution ou certains phénomènes ne pas être représentés à très basse résolution. Le système résolu, ou modèle, n'est alors pas tout à fait le même selon la précision du domaine.
Une dernière possibilité est de prendre

$$\mathbf{A}^c = I_f^c \mathbf{A}^f I_c^f \tag{3.28}$$

(voir Trottenberg [62]), qui est appelé alors l'opérateur de Galerkin de par la formulation en éléments finis dans laquelle cette équation apparaît implicitement. Cette méthode est précisément pratique pour les discrétisations en éléments finis. Elle possède de plus des propriétés particulières (voir Hemker [28], McCormick [43]).
Il suffit de considérer l'évolution du résidu $\mathbf{r} = \mathbf{A}\mathbf{e}$ en utilisant l'évolution de $\mathbf{e}$ (3.23) :

$$\mathbf{r}^f_{\text{after CGC}} = \mathbf{A}^f \mathbf{e}^f_{\text{after CGC}} = \left(\mathbf{A}^f - \mathbf{A}^f I_c^f (\mathbf{A}^c)^{-1} I_f^c \mathbf{A}^f\right) \mathbf{e}^f = \left(\mathbf{I} - \mathbf{A}^f I_c^f (\mathbf{A}^c)^{-1} I_f^c\right) \mathbf{r}^f \tag{3.29}$$

Si on utilise la formulation de Galerkin, on observe alors que le résidu après transfert sur la grille grossière est nul :

$$I_f^c \mathbf{r}^f_{\text{after CGC}} = (\mathbf{I} - \underbrace{I_f^c \mathbf{A}^f I_c^f}_{=\mathbf{A}^c} (\mathbf{A}^c)^{-1}) I_f^c \mathbf{r}^f = 0 \tag{3.30}$$

La formulation de Galerkin permet de nous assurer que le passage par la grille grossière a bien éliminé toutes les basses fréquences du problème fin, et pas seulement du problème grossier.
Les hautes fréquences sont situées dans le noyau de I_f^c, le passage par la grille grossière n'a donc aucun effet sur elles. Si la méthode de relaxation lisse efficacement, alors un cycle complet de multigrilles (lissage + passage par la grille grossière) permettra d'annuler le résidu rapidement si l'opérateur d'interpolation est suffisamment fidèle.

3.2.5 Illustration

Dans notre exemple, le facteur de convergence d'un passage par la grille grossière $\mathbf{K}$ (3.24) est étudié en utilisant l'analyse de Fourier. Le système résolu est l'équation de Poisson (3.3), la

méthode de relaxation utilisée est la méthode de Jacobi pondéré, avec $\omega = 2/3$. Nous utilisons deux maillages de raffinement d'ordre 2, le plus fin ayant une taille de maille $h = \frac{1}{100}$.
Enfin, nous utilisons les opérateurs d'interpolation linéaire et de restriction full-weighting déjà présentés (voir (3.18) et (3.17)).
Nous pouvons exprimer le symbole de l'opérateur discret $\mathbf{A}^f$ selon le mode k et la taille de discrétisation h :

$$\widetilde{\mathbf{A}^f}(m) = \frac{-4}{h^2}\sin^2(\frac{mh}{2}) \tag{3.31}$$

Celui de l'interpolation linéaire est égal à $\widetilde{I_f^c}(m) = 1 - \sin^2(\frac{mh}{2})$, (voir équation (3.19)). L'opérateur de restriction a le même symbole car il est le transposé de l'interpolation, à une constante multiplicative près.
Nous choisissons de construire l'opérateur $\mathbf{A}^c$ de la même façon que $\mathbf{A}^f$ en changeant la valeur de h et en considérant que la maille du domaine Ω^c a une taille $2h$.
On obtient le symbole de $\mathbf{K} = \mathbf{I} - I_c^f(\mathbf{A}^c)^{-I} I_f^c \mathbf{A}^f$:

$$\widetilde{\mathbf{K}}(m) = 1 - \left(1 - \sin^2(\frac{mh}{2})\right)^2 \frac{4\sin^2(\frac{mh}{2})}{\sin^2(mh)} \tag{3.32}$$

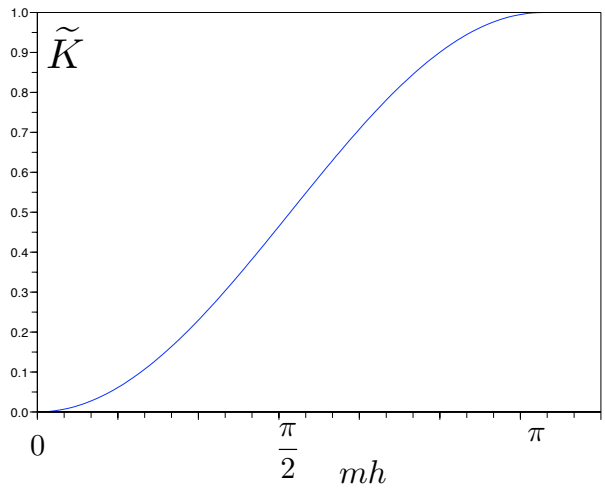

FIG. 3.7 – Le symbole du facteur de convergence de $\mathbf{K}$, en fonction de $mh =]0, \pi[$ avec $h = 1/100$.

Le symbole de ce facteur $\mathbf{K}$ est visible sur la figure 3.7. On observe que son comportement est réellement complémentaire du comportement du symbole de $\mathbf{P}_{\omega=2/3}$; dans le sens où le symbole est proche de 0 pour les basses fréquences et proche de 1, pour les hautes.

On visualise, sur la figure 3.8, la valeur du symbole de $\mathbf{K}_{\text{global}}$ (3.25) pour la résolution de l'équation de Poisson avec une méthode de Jacobi pondérée dans le cas où $\omega = 2/3$ et où $\nu_1 = \nu_2 = 2$. Nous vérifions que les méthodes multigrilles, utilisées avec les bons paramètres, se révèlent très efficaces.
Nous comparons cette valeur avec le symbole $\widetilde{\mathbf{P}_\omega^f}^{(\nu_2+\nu_1)}$. Ceci nous permet de comprendre l'impact de la correction par grille grossière. Si, pour résoudre l'équation résiduelle, on utilise plusieurs

grilles, le calcul du passage par les grilles grossières se fera sans coût supplémentaire. C'est donc une méthode à fort potentiel.

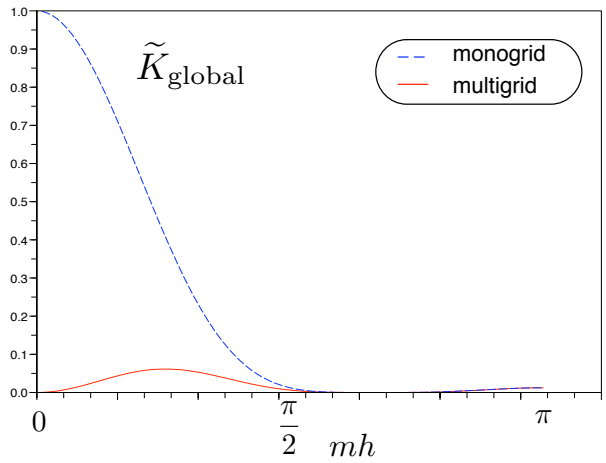

FIG. 3.8 – Le facteur de convergence $\widetilde{\mathbf{K}}_{\text{global}}$, en fonction de $mh =]0, \pi[$; avec $h = 1/100$, en trait plein. En trait discontinu, le facteur de convergence de la même méthode sans passage par la grille grossière.

3.2.6 Résumé

Les méthodes multigrilles reposent sur la vérification de plusieurs points :

1. Trouver, si possible, une discrétisation h-elliptique du système linéaire sur une grille de taille h, sans quoi aucune méthode de relaxation n'éliminera efficacement les hautes fréquences de l'erreur.
2. Choisir un schéma de relaxation avec un bon facteur de convergence pour les hautes fréquences de l'erreur.
3. Choisir des opérateurs de transfert des variables d'une grille à l'autre, dont la somme des ordres seconds est supérieure ou égale à l'ordre de l'opérateur
4. Choisir l'opérateur du système grossier comme une bonne approximation de l'opérateur du système fin.

3.3 Les systèmes non linéaires exposés

De nombreux systèmes modélisent des phénomènes physiques peu ou fortement non-linéaires. Le système originel (3.1) devient alors, dans le cas continu :

$$A(\mathbf{x}_0^a) = \mathbf{g} \tag{3.33}$$

où $A(.)$ est un opérateur non linéaire.
Les méthodes multigrilles précédentes ne sont pas utilisables telles quelles, mais il existe des

adaptations dans le cas non-linéaire.
Afin de bien comprendre les différences des diverses adaptations, nous allons décrire auparavant quelques méthodes classiques de résolution de systèmes non linéaires. Puis nous présenterons les deux algorithmes multigrilles non linéaires, le Newton-Multigrille et le Full Approximation Scheme (ou FAS), inspirés par les méthodes présentées ci -dessous.

3.3.1 Les méthodes de résolution classiques

La méthode de Newton

La plus connue de ces méthodes est sans doute la méthode de Newton. Elle repose sur des linéarisations globales. En effet, il s'agit d'écrire le système A sous forme de série de Taylor autour de l'itéré $\mathbf{x}_{0,k}$:

$$A(\mathbf{x}_{0,k} + \delta\mathbf{x}_0) = A(\mathbf{x}_{0,k}) + \mathbf{A}\delta\mathbf{x}_0 + \text{termes d'ordres supérieurs}$$

avec $\mathbf{A}$, le linéaire tangent de A, qui est aussi la Jacobienne de A.
Si on néglige les termes d'ordres supérieurs, et que l'on soustrait $\mathbf{g}$ aux deux membres de l'équation, on obtient :

$$A(\mathbf{x}_{0,k} + \delta\mathbf{x}_0) - \mathbf{g} = A(\mathbf{x}_{0,k}) - \mathbf{g} + \mathbf{A}\delta\mathbf{x}_0$$

La méthode de Newton utilise cette approximation d'ordre 1 et cherche la correction $\delta\mathbf{x}_0$ qui vérifie $A(\mathbf{x}_{0,k} + \delta\mathbf{x}_0) = \mathbf{g}$. Ce qui revient à résoudre :

$$\mathbf{A}\delta\mathbf{x}_0 = \mathbf{r}_k$$

avec $\mathbf{r}_k$, le résidu $\mathbf{g} - A(\mathbf{x}_{0,k})$.

Autrement dit, la méthode de Newton consiste à résoudre une équation résiduelle supposée linéaire, pour calculer l'itéré suivant.
En voici l'algorithme :

Algorithme de la méthode de Newton :
Données : $\mathbf{x}_{0,k=0}$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. Calcul du résidu :
$$\mathbf{r}_k = \mathbf{g} - A(\mathbf{x}_{0,k})$$
2. Résolution de l'équation résiduelle :
$$\mathbf{A}\delta\mathbf{x}_0 = \mathbf{r}_k, \quad \text{sur } \Omega \tag{3.34}$$
où $\mathbf{A}$ est le linéaire tangent de A autour de $\mathbf{x}_{0,k}$.
3. Corriger la solution,
$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_0 \tag{3.35}$$
4. $k = k + 1$

Fin de la Boucle

Pour étudier la convergence de la méthode, nous reprenons la démonstration de l'article de Yavneh *et al.* [70], en se plaçant dans le cas scalaire où il faut trouver x_0^a réel tel que :

$$A(x_0^a) = g$$

où A est une fonction non linéaire deux fois continue et dérivable. On suppose que $|A'(x)| \geq K > 0, \forall x \in D_\rho$, avec $D_\rho = \{x, |x - x_0^a| < \rho\}$.
Une itération de Newton s'écrit, en partant de $x_{0,k}$, approximation de x_0^a :

$$A'(x_{0,k})(x_{0,k+1} - x_{0,k}) = g - A(x_{0,k})$$

Pour une bonne estimation de l'état optimal, en particulier si $|A''(x)| \leq R, \forall x \in D_\rho$, alors

$$\left|\frac{e_{k+1}}{e_k}\right| \leq \left|\frac{e_k R}{2K}\right|$$

et l'itération converge si $|e_k| < \frac{2K}{R}$.
Le comportement asymptotique est quadratique :

$$\lim_{k\to\infty} \left|\frac{e_{k+1}}{e_k^2}\right| = \left|\frac{A''(x_0^a)}{2A'(x_0^a)}\right|$$

La méthode de Newton a l'avantage d'avoir une convergence quadratique. Cependant son bassin d'attraction est petit, ce qui impose de partir d'une première estimation de la solution suffisamment bonne, pour profiter des bons résultats de convergence.

Les méthodes Quasi-Newton

Dans de nombreux cas de systèmes complexes, il est souvent difficile d'accéder à la Jacobienne de A, et encore plus coûteux de connaître son inverse.
Les méthodes de Quasi-Newton englobent toutes les méthodes qui se proposent d'approcher l'inverse de $\mathbf{A}$ par une matrice $\mathbf{D}$. L'équation résiduelle est alors remplacée par :

$$\delta\mathbf{x}_0 = \mathbf{x}_{0,k+1} - \mathbf{x}_{0,k} = -\mathbf{D}(A(\mathbf{x}_{0,k}) - \mathbf{g}) \tag{3.36}$$

L'algorithme est le même que pour la méthode de Newton, mais on résout l'équation ci-dessus (3.36) au lieu de résoudre l'équation (3.34).
Parmi les plus complexes des méthodes, se trouve la méthode BFGS (Broyden-Fletcher-Goldfarb-Shanno), méthode déjà explicitée section 2.1.2 qui approche la matrice $(\mathbf{A})^{-1}$ en utilisant une formule BFGS à mémoire limitée. Cette dernière, combinée à une recherche linéaire pour le pas de descente, forme la base de la méthode M1qn3, conçue par Gilbert et Le Maréchal, [21].

Dès lors que l'inverse de la Jacobienne est approchée et non pas connue de manière exacte, la convergence quadratique obtenue avec la méthode de Newton, n'est plus valable. Il est prouvé que les méthodes Quasi-Newton convergent au moins linéairement, selon la validité de l'approximation $\mathbf{D}$.

Les méthodes de relaxation non linéaires

Une dernière possibilité est non pas de faire une approximation d'ordre 1, puis une résolution itérative de l'équation résiduelle mais de procéder dans l'ordre inverse. Ces méthodes reposent sur une linéarisation locale, à l'inverse des méthodes type Newton reposant sur une linéarisation globale.
En d'autres termes, nous supposons que l'opérateur A comporte une partie linéaire, et une partie non linéaire $A = \mathbf{A}_1 + A_2$. Illustrons l'exemple sur une itération d'une méthode de relaxation classique comme celle de Jacobi (voir exemple de la section 3.1.1) :

$$\mathbf{M}_1\mathbf{x}_{0,k+1} + A_2(\mathbf{x}_{0,k+1}) = \mathbf{N}_1\mathbf{x}_{0,k} + \mathbf{g}$$

avec $\mathbf{M}_1$ la diagonale de $\mathbf{A}_1$ et $\mathbf{N}_1 = \mathbf{M}_1 - \mathbf{A}_1$.
Ce n'est qu'après cette opération que sera utilisée une méthode de Newton pour estimer $A_2(\mathbf{x}_{0,k+1})$ en introduisant $\mathbf{A}_2$ la Jacobienne de A_2 autour de $\mathbf{x}_{0,k}$.
Ce type de méthode est appelé Jacobi-Newton, alors qu'utiliser une méthode de Newton puis Jacobi pourrait s'appeler Newton-Jacobi.

Dans les sous-sections suivantes, nous présentons les méthodes multigrilles adaptées au cas non linéaire. Pour construire les différents algorithmes, l'inspiration est venue des méthodes présentées ci-dessus.

3.3.2 La méthode Newton-multigrille (N-MG)

La méthode, appelée Newton-multigrille, que l'on notera par la suite, N-MG, est l'application des méthodes multigrilles à la méthode de Newton.

Présentation

Il s'agit d'appliquer la méthode de Newton et de résoudre les différents systèmes linéarisés en utilisant une méthode multigrille.
Nous remplaçons donc la résolution directe de l'équation résiduelle (3.34), dans l'algorithme de la méthode de Newton, par l'algorithme de la méthode multigrille décrit section 3.1.3. L'algorithme N-MG présente ainsi deux boucles, une boucle externe dans laquelle on met à jour le résidu pour définir la nouvelle équation résiduelle et une boucle interne dans laquelle on résout l'équation résiduelle en lui appliquant une méthode multigrille.

Cet algorithme permet d'utiliser les outils de calculs linéaires plus fiables que les calculs non linéaires.

Propriétés de convergence

La convergence de la méthode Newton-Multigrille dépend de la convergence de la méthode de Newton.
Si l'on est déjà proche de la solution, la convergence de la méthode N-MG est au mieux la convergence quadratique de la méthode de Newton. Sinon, la convergence sera faible jusqu'à atteindre le bassin d'attraction. Il se peut également qu'il y ait convergence vers un minimum local.

Au sein de chaque boucle externe, la résolution du problème linéaire dépend des paramètres

des méthodes multigrilles dont la convergence a été étudiée dans la section précédente.

En ce qui concerne les critères d'utilisation, en plus de devoir régler les paramètres de la méthode multigrille, il nous faut maintenant choisir combien de cycles de multigrille effectuer au sein de chaque boucle interne.
Dans le livre de Trottenberg [62], sont comparés sur un exemple, le fait de doubler le nombre de cycles par boucle externe au fait de garder un nombre de cycle constant. L'efficacité, c'est-à-dire, la précision du résultat après le même nombre total de cycles de multigrille, est équivalente pour les deux méthodes. Dans ce cas précis, la sensibilité de ce paramètre n'est donc pas prouvée et est difficile à étudier.

3.3.3 L'algorithme Full Approximation Scheme (FAS)

L'algorithme Full Approximation Scheme, que nous nommerons FAS, est ce qu'on pourrait appeler la méthode Multigrille-Newton pour les méthodes de relaxation non linéaires. C'est-à-dire qu'au lieu de faire l'hypothèse d'approximation linéaire de la méthode de Newton puis d'utiliser les méthodes multigrilles, nous allons dérouler l'algorithme des méthodes multigrilles pour le système non-linéaire et ne faire l'hypothèse de linéarité que lorsqu'on ne peut plus faire autrement. Voilà pourquoi, dans quelques articles, la méthode FAS est incluse dans les méthodes de linéarisation locale, alors que la méthode N-MG est considérée comme une méthode de linéarisation globale.

Principe

La méthode FAS est une généralisation de l'algorithme du cas linéaire au non linéaire. Nous l'illustrons dans le cas de 2 grilles, suivant les mêmes notations que dans le cas linéaire, avec un opérateur non linéaire $A(.)$ au lieu de $\mathbf{A}$. On peut, bien sûr là aussi, généraliser par méthode récursive cet algorithme à plus de deux grilles.
La différence avec le cas linéaire réside dans le fait que l'on ne travaille pas directement avec les erreurs mais avec les approximations complètes de la solution, d'où le nom de la méthode.
En effet, dans le cas non linéaire l'équation résiduelle à résoudre sur la grille grossière Ω^c est différente :

$$A(\mathbf{x}_0 + \mathbf{e}) - A(\mathbf{x}_0) = \mathbf{r} = \mathbf{g} - A(\mathbf{x}_0)$$

Il nous faut transposer $\mathbf{r}$ et $\mathbf{x}_0$, et non plus seulement le résidu. Nous définissons $\hat{I}_f^c$ comme l'opérateur de transfert de l'approximation de la solution $\mathbf{x}_0$ de la grille Ω^f à Ω^c. En effet cet opérateur peut être différent de celui transposant le résidu $\mathbf{r}$ car n'évoluant pas dans le même espace, $\mathbf{r}$ et $\mathbf{x}$ n'ont pas les mêmes caractéristiques.

Algorithme de la méthode FAS (voir schéma 3.9) :
Données : $\mathbf{x}_{0,k=0}^f$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. Relaxation : appliquer ν_1 fois la méthode de relaxation notée SMOOTH à $\mathbf{x}_0^f$:

$$\mathbf{x}_0^f = SMOOTH^{\nu_1}(\mathbf{x}_{0,k}^f, A^f, \mathbf{g}^f)$$

2. Correction par grille grossière :

(a) Calculer le nouveau second-membre :

$$\mathbf{g}^c = \mathbf{r}^c + A^c(\hat{I}_f^c \mathbf{x}_0^f)$$

avec $\mathbf{r}^c$ le résidu de la grille fine transféré à basse résolution : $\mathbf{r}^c = I_f^c \mathbf{r}^f = I_f^c(\mathbf{g}^f - A^f(\mathbf{x}_0^f))$

(b) Résoudre, par une méthode de relaxation ou un solveur direct :

$$A^c(\mathbf{x}_0^c) = \mathbf{g}^c, \quad \text{sur } \Omega^c \tag{3.37}$$

(c) Calculer la correction $\delta \mathbf{x}_0^c = \mathbf{x}_0^c - \hat{I}_f^c \mathbf{x}_0^f$ et en déduire une approximation de la solution sur Ω^f,

$$\mathbf{x}_{0,\text{ after CGC}}^f = \mathbf{x}_0^f + I_c^f \delta \mathbf{x}_0^c \tag{3.38}$$

3. Relaxation : appliquer ν_2 fois la méthode de relaxation pour trouver le nouvel $\mathbf{x}_0^f$.

$$\mathbf{x}_{0,k+1}^f = SMOOTH^{\nu_2}(\mathbf{x}_{0,\text{ after CGC}}^f, A^f, \mathbf{g}^f)$$

4. $k = k + 1$

Fin de la Boucle

SMOOTH indique maintenant une méthode de résolution de systèmes non linéaires.

La dernière étape de lissage permet, comme dans le cas linéaire, de réduire encore les hautes fréquences restantes de l'erreur de la grille fine et celles introduites par l'interpolation. Elle permet aussi de résoudre les interactions d'échelles dues aux non-linéarités.

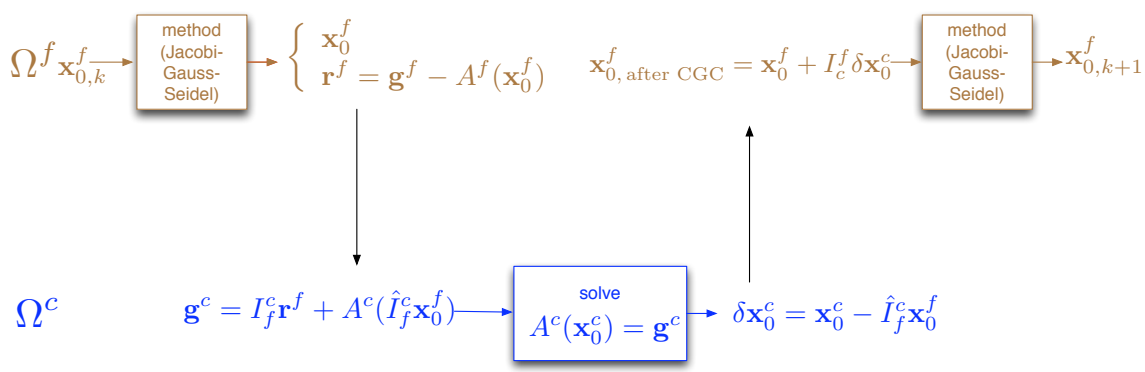

FIG. 3.9 – L'algorithme FAS, les méthodes multigrilles pour la résolution de systèmes non linéaires

Remarque importante :
Si on applique une correction par grille grossière à la solution $\mathbf{x}_0^{f,a}$ de la grille fine, on transfère alors un résidu nul et l'équation résiduelle de la grille grossière devient :

$$A^c(\mathbf{x}_0^c) = A^c(\hat{I}_f^c \mathbf{x}_0^{f,a}).$$

La solution est clairement $\hat{I}_f^c \mathbf{x}_0^{f,a}$, ce qui donne une correction $\delta \mathbf{x}_0^c$ nulle.
On démontre par là, une propriété importante du FAS, à savoir que si la solution du système a été trouvée sur la grille fine alors la correction par grille grossière ne la modifie pas.

Dans la suite du manuscrit, on fera référence à l'équation résiduelle (3.37) en l'écrivant sous la forme :

$$A_{\text{FAS}}^c(\mathbf{x}_0^c) = I_f^c \mathbf{g}^f \tag{3.39}$$

si on définit l'opérateur A_{FAS}^c ainsi :

$$A_{\text{FAS}}^c(\mathbf{x}_0^c) = A^c(\mathbf{x}_0^c) + I_f^c A^f(\mathbf{x}_0^f) - A^c(\hat{I}_f^c \mathbf{x}_0^f) \tag{3.40}$$

L'algorithme FAS étant une extension du cas linéaire, on peut l'appliquer à des systèmes linéaires, il aura le même effet que l'algorithme classique.

Propriétés de convergence

Comme démontré dans l'article de Yavneh *et al.* [70], on se place dans le cas scalaire où il faut trouver x_0^a réel tel que :

$$A(x_0^a) = g$$

où A est une fonction non linéaire deux fois continue et dérivable. On rappelle que l'on suppose $|A'(x)| \geq K > 0, \forall x \in D_\rho$, avec $D_\rho = \{x, |x - x_0^a| < \rho\}$.
On considère qu'il existe une fonction non linéaire facile à inverser $\hat{A}$ qui est une approximation de A, et que $x_{0,k}$ est l'approximation de x_0^a. Alors la résolution du problème approché s'écrit :

$$\hat{A}(x_{0,k+1}) - \hat{A}(x_{0,k}) = g - A(x_{0,k}) \tag{3.41}$$

On ne distingue pas x_0^f de x_0^c car ce n'est plus qu'un scalaire, mais $\hat{A}$ peut être assimilé à A^c (ou à $I_c^f A^c I_f^c$).

Supposons que $\hat{A}$ approche A à l'ordre ϵ avec un ϵ petit :

$$\hat{A}(x) = A(x) + \epsilon\phi(x) \tag{3.42}$$

avec ϕ une fonction deux fois continue et dérivable.
En remplaçant (3.42) dans (3.41) :

$$\epsilon\phi(x_{0,k+1}) - \epsilon\phi(x_{0,k}) = g - A(x_{0,k+1}) = A(v_0) - A(x_{0,k+1})$$

car v_0 est solution du système.
En posant $e_k = x_{0,k} - x_0^a$, nous utilisons le théorème de la moyenne pour écrire :

$$\epsilon\phi'(\chi_1)(e_{k+1} - e_k) = -A'(\chi_2)e_{k+1}$$

où $\chi_1 \in (x_{0,k+1}; x_{0,k})$ et $\chi_2 \in (x_0^a; x_{0,k+1})$.
On suppose $|\phi'(x)| \leq P < K/2\epsilon, \quad \forall x \in D_\rho$. Alors, dans ce cas, on peut écrire :

$$\left|\frac{e_{k+1}}{e_k}\right| = \left|\frac{\epsilon\phi'(\chi_1)}{A'(\chi_2) + \epsilon\phi'(\chi_1)}\right| \leq \left|\frac{\epsilon P}{K - \epsilon P}\right| < 1$$

Il y a donc convergence vers la solution x_0^a et le comportement asymptotique est

$$\lim_{k\to\infty}\left|\frac{e_{k+1}}{e_k}\right| = \left|\frac{\epsilon\phi'(x_0^a)}{A'(x_0^a)+\epsilon\phi'(x_0^a)}\right| = \epsilon\left|\frac{\phi'(x_0^a)}{A'(x_0^a)}\right| + O(\epsilon^2)$$

La convergence de la méthode est donc linéaire et dépend de la bonne approximation de A.

Contrairement à la méthode N-MG, la convergence quadratique n'est pas assurée. Cependant, le FAS peut être plus efficace que le N-MG car sa dépendance à la qualité de la première estimation est moindre.
Dans un cas plus général, on peut extrapoler les résultats précédents aux opérateurs A^f et A^c (3.27). Les résultats de convergence du FAS dépendent fortement de la qualité d'approximation de l'opérateur grossier A^c. On peut espérer une bonne approximation si cet opérateur est construit de manière à approcher la relation de Galerkin :

$$\|A^c - I_f^c A^f I_c^f\| \leq (\Delta x^f)^\alpha, \quad \alpha > 0 \tag{3.43}$$

S'il est difficile de vérifier cette condition de convergence, on peut utiliser comme Nash [40] ou Ta'asan [59] une recherche linéaire pour trouver un pas α à appliquer à la correction par grille grossière :

$$\mathbf{x}^f_{0,\text{ after CGC}} = \mathbf{x}^f_{0,k} + \alpha I_c^f \delta\mathbf{x}_0^c \tag{3.44}$$

et contraindre ainsi la correction par grille grossière à diminuer l'erreur.

3.3.4 Confrontation et hybridation

Pour Trottenberg [62], les deux méthodes peuvent être équivalentes en résultats et en temps de convergence. Mais pour certains problèmes plus complexes (voir Yavney et Dardyck [70]), elles se comportent différemment et le choix de la méthode devient réellement non anodin. L'article compare les méthodes sur différents tests créés en modifiant les paramètres d'une équation de diffusion non linéaire, 1D puis 2D, qui modélise la conductivité hydraulique des sols non saturés. On observe que la convergence de l'algorithme FAS est moins dépendante des paramètres du modèle que la méthode N-MG. Plus les non-linéarités sont fortes, plus le N-MG est long à atteindre le taux de convergence quadratique.

De nombreuses autres méthodes adaptées aux non linéarités ont été créées.
En 1985, Hackbusch [26] a développé en parallèle à la méthode FAS, une méthode multigrille non linéaire sous l'acronyme NMGM. Et de nouvelles méthodes voient encore le jour. Par exemple, en 2006, Yavney et Dardyck [70] ont proposé une hybridation de N-MG et FAS, nommée Multilevel Nonlinear Method (MNM) ; qui consiste à appliquer l'algorithme FAS sur la partie non linéaire du système, supposée petite, et le Newton-Multigrille sur la partie linéaire. L'article compare les méthodes sur différents tests créés en modifiant les paramètres d'une équation de diffusion non linéaire, 1D puis 2D. La convergence de MNM est en général meilleure que celle du FAS, mais les résultats dépendent fortement des paramètres du modèle.
On peut aussi imaginer d'autres méthodes plus simples à implémenter : un algorithme FMG-FAS dans lequel, on commence par résoudre l'équation sur la grille grossière avec une approximation linéaire. À chaque transfert sur une grille plus précise, on ajoute une part de non-linéarité jusqu'à arriver à la grille la plus fine et à l'opérateur non linéaire sans approximation.
Ce préconditionnement par la grille grossière est censé fortement accélerer la convergence à moindre coût. On effectue ensuite les V-cycles ou W-cycles traditionnels.

Deuxième partie

Les méthodes multigrilles pour l'assimilation de données

Les méthodes multigrilles, qui ont été présentées au chapitre précédent, ont la particularité de corriger certains défauts des méthodes de relaxation. Les méthodes de relaxation, ainsi parées, sont aussi performantes que les méthodes directes, voire plus car elles ont un champ d'applications plus large. En effet, dans le cas de systèmes complexes de grande taille, les méthodes directes sont inutilisables car elles nécessitent trop de place mémoire et engendrent des opérations coûteuses.

Dans l'histoire de l'assimilation de données, que nous avons introduit dans le premier chapitre, se dégage l'assimilation de données variationnelle. Elle repose sur les méthodes de contrôle optimal et consiste à minimiser une fonction qui mesure l'écart entre la solution du modèle et les observations disponibles. Nous expliquons dans le deuxième chapitre que cette méthode a une résolution très coûteuse dans le cadre de la modélisation en géophysique. De plus, la méthode incrémentale réduite, introduite pour diminuer le coût de calcul, ne donne pas de résultats satisfaisants ; même si elle est couplée avec un préconditionnement de la fonction coût.
Or, nous l'avons vu, les méthodes variationnelles d'assimilation de données sont résolues par méthodes itératives. En effet, la minimisation s'obtient par itérations successives, chaque itération comprenant une intégration du modèle suivie d'une intégration rétrograde du modèle adjoint. Notre objectif est donc d'étudier les méthodes multigrilles comme une nouvelle astuce pour alléger le coût de calcul d'un problème d'assimilation de données.

Nous basons notre travail sur les applications déjà faites en contrôle optimal (Ta'asan [58], Nash [48], [40]). Si nous considérons que $\mathbf{x}_0$ est le vecteur de contrôle, assimiler revient à résoudre l'équation d'Euler $\nabla_{\mathbf{x}_0}\mathcal{J} = 0$ qui décrit la condition nécessaire d'optimalité.
Nous nous intéressons, dans le chapitre suivant, à l'assimilation sur un modèle linéaire, à une dimension. La linéarité nous permet de nous placer dans un contexte particulier où l'étude théorique est pertinente. La fonction coût utilisée comporte un terme de régularisation, dont l'importance a été relevée dans le chapitre 2. En utilisant l'analyse de Fourier, nous étudions l'influence de la diffusion numérique du modèle sur l'ellipticité du système et l'effet du terme de régularisation. Nous étudions également l'effet du préconditionnement sur la convergence de la méthode.

Au chapitre 5, le modèle devient non-linéaire. Nous orientons l'étude vers la comparaison entre les méthodes incrémentales, reposant sur une linéarisation globale, et les méthodes de résolution non-linéaires. En particulier, nous présentons la méthode multi-incrémentale, l'algorithme Gauss-Newton Multigrille (GN-MG), application directe des méthodes multigrilles à la méthode incrémentale ; et l'algorithme Full-Approximation Scheme (FAS).
Nous testons la robustesse de ces différents algorithmes, en estimant les erreurs de linéarisation, et d'approximation de la grille grossière ; ainsi qu'en faisant varier les différents paramètres des méthodes. Nous abordons également l'effet du préconditionnement.

Enfin, le chapitre 6 est l'occasion d'aller plus loin dans les modèles non-linéaires, en adaptant nos travaux à un modèle Shallow-Water, le plus simple des modèles océans. Sur ce modèle, nous mettons en évidence des difficultés de convergence. Celle-ci se trouve étroitement liée à la distance entre l'ébauche et l'état optimal ; ainsi qu'à celle entre les opérateurs fins et grossiers. Nous présentons différentes possibilités pour minimiser ces distances. Les algorithmes multigrilles se trouvent alors plus performants que ce soient avec ou sans préconditionnement.

4

LES MÉTHODES MULTIGRILLES POUR L'ASSIMILATION DE DONNÉES VARIATIONNELLE

Sommaire

4.1 L'algorithme multigrille adapté au contrôle optimal 82
4.1.1 L'algorithme 82
4.1.2 Rappel des conditions d'utilisation 84
4.2 En ce qui concerne l'assimilation de données 84
4.2.1 L'ellipticité du système 84
4.2.2 La méthode d'optimisation 85
4.2.3 Étude de la convergence 86
4.3 Application à une équation d'advection linéaire 87
4.3.1 Présentation de la configuration 87
4.3.2 La propriété de lissage 89
4.3.3 Étude de la convergence 96
4.4 Conclusion 105

Ce chapitre expose l'application des méthodes multigrilles, présentées au chapitre 3 au problème d'assimilation de données variationnelle, décrit aux chapitres 1 et 2.
Nous appliquons les méthodes multigrilles dans le cadre d'un problème d'assimilation d'un modèle linéaire. Cette illustration sur un cas linéaire est d'autant plus intéressante que bien souvent pour résoudre des systèmes non linéaires, on linéarise localement le système pour se ramener à la résolution de plusieurs systèmes linéaires, qui sont les linéaires tangents du système originel. C'est le cas de la méthode incrémentale pour l'assimilation de données (voir section 2.2).
Nous étudions, sur ce cas linéaire, le principe de fonctionnement des méthodes multigrilles. Notamment, nous nous intéressons à l'influence des différents paramètres du modèle et de la fonction coût sur la convergence de la méthode. Nous testons également l'effet du préconditionnement.

4.1 L'algorithme multigrille adapté au contrôle optimal

Nous rappelons que l'assimilation variationnelle de données consiste à estimer les paramètres d'un modèle afin de rendre les résultats plus proches des observations. L'estimation de ces paramètres se fait au travers de la minimisation d'une fonction coût $\mathcal{J}$ comme celle décrite équation (1.1) :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{4.1}$$

dans le cas d'un modèle et d'un opérateur d'observation linéaires. Nous rappelons les significations des différents symboles :
$\mathbf{x}_0$: état initial du système
$\mathbf{M}_{0,i}$: modèle, supposé ici linéaire, décrivant l'évolution temporelle de $\mathbf{x}$.
$\mathbf{H}_{t_i}$: opérateur, supposé ici linéaire, qui transfère l'état $\mathbf{x}$ à l'espace des observations.
$\mathbf{y}_{t_i}$: les observations réelles (signaux physiques)
$\mathbf{x}^b$: ébauche de l'état initial, estimation de l'état initial
$\mathbf{R}_i$: matrice de covariance d'erreur de l'observation $\mathbf{y}_i$
$\mathbf{B}$: matrice de covariance d'erreur de l'ébauche.

Nous appliquons les méthodes multigrilles à ce problème d'optimisation. On suppose que l'on connaît une première estimation de cet état, l'ébauche $\mathbf{x}^b$. On va chercher la meilleure estimation $\mathbf{x}_0^a$ selon les statistiques d'erreur supposées connues en résolvant

$$\mathcal{J}(\mathbf{x}_0^a) = \min_{\mathbf{x}_0}\mathcal{J}(\mathbf{x}_0)$$

4.1.1 L'algorithme

Notre but est d'adapter les méthodes multigrilles au problème d'assimilation de données décrit précédemment et en particulier à la résolution du système d'optimisation. Dans ce chapitre, nous considérons un modèle et un opérateur d'observation linéaires, la fonction à minimiser est donc celle de l'équation (4.1).
Une condition nécessaire pour que $\mathbf{x}_0^a$ soit un minimum de $\mathcal{J}(\mathbf{x}_0)$ est l'équation d'Euler :

$$\nabla\mathcal{J}(\mathbf{x}_0^a) = 0. \tag{4.2}$$

C'est à ce système que nous allons appliquer l'algorithme multigrille, en se basant sur les travaux de Nash [48], et Ta'asan [59] qui ont étudié des exemples d'applications des multigrilles au contrôle optimal. On considère l'équation (3.1) en remplaçant $\mathbf{A}\mathbf{x}_0 - \mathbf{g}$ par $\nabla\mathcal{J}(\mathbf{x}_0)$.

On reprend les mêmes notations que dans le chapitre sur les méthodes multigrilles. L'exposant c (resp. f) désigne les variables de la grille grossière (resp. fine). On note donc $\nabla\mathcal{J}^c$ et $\nabla\mathcal{J}^f$ les discrétisations de $\nabla\mathcal{J}$. Pour échanger les informations entre les deux grilles, on utilise un opérateur de restriction I_c^f et un opérateur d'interpolation I_f^c.

Sur la grille fine Ω^f, on résout :

$$\nabla\mathcal{J}^f(\mathbf{x}_0^f) = 0,$$

Pour des systèmes linéaires, sur la grille à plus basse résolution Ω^c, l'équation résolue est alors l'équation résiduelle (voir équation (3.11)) :

$$\mathbf{A}^c \delta \mathbf{x}_0 = \mathbf{r}^c$$

avec $\mathbf{r}^c = I_f^c \mathbf{r}^f = I_f^c(\mathbf{g}^f - \mathbf{A}^f \mathbf{x}_{0,k}^f)$.
Le système optimal de l'assimilation de données, utilisé avec un modèle linéaire, donne un gradient affine et un hessien $\mathcal{H} = \nabla^2 \mathcal{J}$ linéaire. Sur la grille grossière, le système à résoudre est donc le suivant :

$$\nabla \mathcal{J}^c(\mathbf{x}_0^c - I_f^c \mathbf{x}_0^f) = \mathcal{H}^c \delta \mathbf{x}_0^c = -I_f^c \nabla \mathcal{J}^f(\mathbf{x}_0^f) \tag{4.3}$$

$\mathcal{H}$ est donc l'équivalent de l'opérateur $\mathbf{A}$ des sections 3.1 et 3.2 sur les méthodes multigrilles.

On obtient l'algorithme des méthodes multigrilles suivant, illustré figure 4.1 :

Algorithme des méthodes multigrilles pour l'assimilation linéaire :
Données : $\mathbf{x}_{0,k=0}^f$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. Relaxation : appliquer ν_1 fois la méthode de relaxation notée SMOOTH à $\mathbf{x}_0^f$:

$$\mathbf{x}_0^f = SMOOTH^{\nu_1}(\mathbf{x}_{0,k}^f, \nabla \mathcal{J}^f)$$

2. Correction par grille grossière :
   (a) Résoudre, par une méthode de relaxation ou un solveur direct :

$$\mathcal{H}^c \delta \mathbf{x}_0^c = -I_f^c \nabla \mathcal{J}^f(\mathbf{x}_0^f), \quad \text{sur } \Omega^c \tag{4.4}$$

   (b) Interpoler la correction et en déduire une approximation de la solution sur Ω^f,

$$\mathbf{x}_{0,\text{afterCGC}}^f = \mathbf{x}_0^f + I_c^f \delta \mathbf{x}_0^c \tag{4.5}$$

3. Relaxation : appliquer ν_2 fois la méthode de relaxation pour trouver le nouvel $\mathbf{x}_0^f$.

$$\mathbf{x}_{0,k+1}^f = SMOOTH^{\nu_2}(\mathbf{x}_{0,\text{afterCGC}}^f, \nabla \mathcal{J}^f)$$

4. $k = k + 1$

Fin de la Boucle

Nous exprimons la correction grille grossière $\delta \mathbf{x}_0^c$ de l'équation (4.5) par l'expression obtenue après résolution de l'équation (4.4) :

$$\mathbf{x}_{0,\text{afterCGC}}^f = \mathbf{x}_0^f - I_c^f (\mathcal{H}^c)^{-1} I_f^c \nabla \mathcal{J}^f(\mathbf{x}_0^f)$$

Un passage par la grille grossière est donc équivalent à une itération d'une méthode de Quasi-Newton qui approcherait l'inverse du hessien haute résolution $(\mathcal{H}^f)^{-1}$ en utilisant le hessien à basse résolution $I_c^f (\mathcal{H}^c)^{-1} I_f^c$.

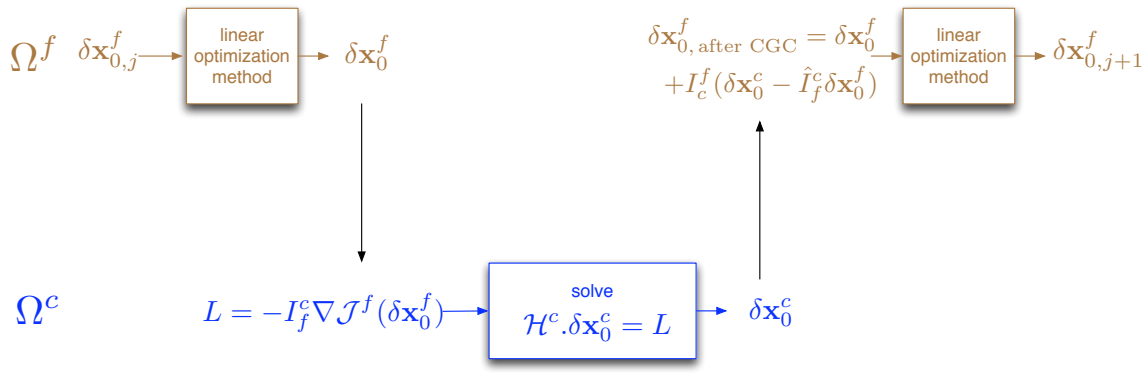

FIG. 4.1 – L'algorithme multigrille pour l'assimilation de données, cas linéaire

4.1.2 Rappel des conditions d'utilisation

L'efficacité des méthodes multigrilles nécessite la vérification de quelques conditions d'utilisation, déjà expliquées section 3.2.
Tout d'abord, l'opérateur du système linéaire de l'équation résiduelle (voir équation (4.3)), doit être elliptique. Il s'agit donc, ici, de vérifier l'ellipticité du Hessien d'un système optimal. L'ellipticité rend possible l'existence d'une méthode itérative éliminant plus vite les hautes fréquences de $\delta\mathbf{x}_0$ que les basses. C'est ce que l'on appelle la propriété de lissage (3.26). Les méthodes multigrilles sont alors assurées d'être plus efficaces que les méthodes monorésolution, à condition qu'elles convergent.
Et pour que la correction par grille grossière aide à la convergence, les opérateurs de transfert doivent être d'ordres suffisamment élevés pour vérifier la relation (3.20). Plus simplement, il suffit d'étudier l'effet d'un passage par la grille grossière à l'aide de son facteur de convergence (3.24). Nous obtenons alors le comportement global de l'algorithme en adaptant le théorème 3.2.1 et la propriété (3.27).
Dans un cas de système optimal, on remplace $\mathbf{A}$ par le Hessien $\mathcal{H}$. La condition de convergence s'écrit alors en fonction des Hessiens de la grille fine et de la grille grossière, et des opérateurs de transfert. Le facteur de convergence est donc égal à

$$\mathbf{K} = \mathbf{I} - I^f_c(\mathcal{H}^c)^{-1}I^c_f\mathcal{H}^f \tag{4.6}$$

De plus, la propriété d'approximation (3.27) est vérifiée si :

$$\|\mathcal{H}^c - I^c_f\mathcal{H}^f I^f_c\| \leq (\Delta x^f)^{\alpha'} \tag{4.7}$$

avec Δx^f le pas de discrétisation en espace de la grille fine, et $\alpha' \geq \alpha$ déterminé par la propriété de lissage (3.26). Le plus souvent, α' est supérieur à l'ordre de consistance des discrétisations et est donc plus grand que α.

4.2 En ce qui concerne l'assimilation de données

4.2.1 L'ellipticité du système

L'équation résiduelle (4.3) fait apparaître le Hessien de la fonction coût. Si on reprend les données sur la fonction coût (4.1), alors on peut calculer, et le gradient, et le Hessien.

Nous obtenons à partir de l'expression de la fonction coût (4.1) :

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

l'expression du gradient :

$$\nabla\mathcal{J}_{\delta\mathbf{x}_0}(\mathbf{x}_0) = \sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0]^T\mathbf{R}_i^{-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_{t_i}] + [\delta\mathbf{x}_0]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{4.8}$$

Le Hessien $\mathcal{H}$ se calcule directement à partir de l'équation précédente :

$$\mathcal{H} = \sum_i (\mathbf{H}_{t_i}\mathbf{M}_{0,i})^T\ \mathbf{R}_i{}^{-1}\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1} \tag{4.9}$$

L'ellipticité vient typiquement de la spécification de $\mathbf{B}^{-1}$ et de $\mathbf{R}^{-1}$. Il nous sera donné de mesurer plus en détails l'importance de ces matrices, et plus particulièrement de la matrice de covariance d'erreur de l'ébauche, dans le cas d'un modèle d'advection linéaire, par la suite dans ce chapitre.

4.2.2 La méthode d'optimisation

Le rôle des méthodes multigrilles est d'accélérer la convergence des basses fréquences de l'erreur, dans le cas où les méthodes itératives réduisent plus efficacement les hautes fréquences. Nous nous intéressons ici à la méthode d'optimisation utilisée pour résoudre le système optimal de l'assimilation de données. L'intérêt des méthodes multigrilles est grandissant si ces méthodes de minimisation permettent de résoudre plus rapidement les hautes fréquences de l'erreur que les basses.

Pour résoudre le système d'optimisation de l'assimilation de données, on choisit d'utiliser la méthode du gradient conjugué, efficace et simple pour des systèmes linéaires aux matrices définies positives (voir section 2.1.2 du chapitre 2) ; ce qui est le cas ici puisque le modèle et l'opérateur d'observation sont linéaires.

La méthode du gradient conjugué repose sur la recherche de directions successives permettant d'atteindre la solution exacte du système étudié. L'algorithme de la méthode a été présenté section 2.1.2. Les méthodes de gradient conjugué, de par leur fonctionnement, recherchent un minimum de proche en proche en ajoutant un vecteur de Krylov. Ces vecteurs représentent des directions de descente et sont associés à des puissances du Hessien $\mathcal{H}$ (ou de $\nabla^2\mathcal{J}$). Il s'agit ainsi d'ajouter à chaque itéré un vecteur de différentiation d'ordre de plus en plus élevé. Un opérateur de différentiation d'ordre élevé, comme le Laplacien et dans un cadre général pour des opérateurs elliptiques, ajoute un terme d'ordre élevé fréquentiellement (voir Nash [40]). La correction calculée par la méthode du gradient conjugué est donc composée, au fur et à mesure des itérations, de plus en plus de hautes fréquences. Les méthodes de gradient conjugué semblent plus à même de corriger les hautes fréquences de l'erreur que les basses.

En conclusion, même si le comportement de la méthode du gradient conjugué est difficile à étudier fréquentiellement (en particulier si on l'étend aux cas non linéaires) et même s'il n'est pas prouvé qu'elle vérifie la propriété de lissage, cette mise en exergue de son fonctionnement justifie l'utilisation des méthodes multigrilles. L'amélioration de la résolution de l'assimilation sur une seule grille semble possible en utilisant une méthode multigrille composée d'une méthode de gradient conjugué.

4.2.3 Étude de la convergence

La convergence des méthodes multigrilles dépend de la méthode d'optimisation et des opérateurs utilisés. Nous supposons que la méthode d'optimisation, couplée au système à résoudre, possède la propriété de lissage (3.26) pour nous concentrer sur l'effet de la correction par grille grossière. Les opérateurs de transfert et les opérateurs discrétisés à basse résolution doivent vérifier la propriété d'approximation.
Celle-ci est satisfaite si l'opérateur grossier approche la relation de Galerkin (4.7) :

$$\|\mathcal{H}^c - I_f^c \mathcal{H}^f I_c^f\| \leq (\Delta x^f)^\alpha \tag{4.10}$$

avec α donnée par la propriété de lissage (3.26).
Elle fait apparaître le Hessien dont l'expression a déjà été calculée équation (4.9) :

$$\mathcal{H} = \sum_i (\mathbf{H}_{t_i}\mathbf{M}_{0,i})^T \ \mathbf{R}_i{}^{-1}\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1} \tag{4.11}$$

Nous énonçons le théorème suivant :

Théorème 4.2.1. *Soit un opérateur $\mathcal{H}$ défini selon le problème de l'assimilation de données (4.11). Soient I_f^c et I_c^f les opérateurs de restriction et d'interpolation entre la grille Ω^f et la grille Ω^c à plus basse résolution.*
Si les opérateurs de transfert sont adjoints entre eux (si il existe $c > 0$ tel que $I_f^c = (I_c^f)^* = \frac{1}{c}(I_c^f)^T$), et si les différents opérateurs composant $\mathcal{H}$ vérifient :

$$\begin{cases} \mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c = \mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f I_c^f \\ (\mathbf{R}^c)^{-1} = \frac{1}{c}(\mathbf{R}^f)^{-1} \\ (\mathbf{B}^c)^{-1} = I_f^c(\mathbf{B}^f)^{-1}I_c^f \end{cases}$$

alors l'opérateur $\mathcal{H}$ vérifie la relation de Galerkin $\mathcal{H}^c = I_f^c\mathcal{H}^f I_c^f$.

La constante c est, dans le cas interpolation linéaire/full-weighting, égale à 2.
Preuve :
Nous calculons la transposée $\mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c$:

$$(\mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c)^T = \left(\mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f I_c^f\right)^T$$

qui devient :

$$(\mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c)^T = \left(I_c^f\right)^T \left(\mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f\right)^T = c\, I_f^c \left(\mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f\right)^T$$

Ce qui nous donne, en utilisant l'expression de $\mathbf{B}^{-1}$ et de $\mathbf{R}^{-1}$ ainsi que les deux expressions précédentes :

$$\sum_i \left(\mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c\right)^T \ \mathbf{R}_i{}^{-1}\mathbf{H}_{t_i}^c\mathbf{M}_{0,i}^c + \mathbf{B}^{c-1} = \sum_i I_f^c \left(\mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f\right)^T \ \mathbf{R}_i{}^{-1}\mathbf{H}_{t_i}^f\mathbf{M}_{0,i}^f I_c^f + I_f^c\mathbf{B}^{f^{-1}}I_c^f$$

Nous retrouvons les expressions des Hessiens (4.11) :

$$\mathcal{H}^c = I_f^c \mathcal{H}^f I_c^f \quad \blacksquare$$

Pour satisfaire ces relations de manière exacte, il faudrait soit transférer les variables sur la grille à haute résolution et faire les calculs directement sur cette grille, soit construire les opérateurs discrets grossiers qui vérifient les relations. L'ennui est que les opérateurs grossiers ainsi générés peuvent être plus complexes que ceux d'origine. Ceci est particulièrement vrai pour l'opérateur modèle. S'il peut être envisageable de tenter de vérifier la deuxième et la troisième égalité pour les matrices de covariance, nous ne chercherons pas à vérifier celle concernant le modèle.
En pratique, on s'attachera plutôt à vérifier une consistance dans les approximations afin que $\mathcal{H}^c$ soit proche de $I_f^c \mathcal{H}^f I_c^f$. La propriété d'approximation s'exprime alors sous forme d'inégalités :

$$\begin{cases} \|\mathbf{H}_{t_i}^c \mathbf{M}_{0,i}^c - \mathbf{H}_{t_i}^f \mathbf{M}_{0,i}^f I_c^f\| \leq (\Delta x^f)^\alpha, \quad \forall i \\ \|(\mathbf{R}^c)^{-1} - \frac{1}{c}(\mathbf{R}^f)^{-1}\| \leq (\Delta x^f)^\alpha \\ \|(\mathbf{B}^c)^{-1} - I_f^c (\mathbf{B}^f)^{-1} I_c^f\| \leq (\Delta x^f)^\alpha \end{cases} \tag{4.12}$$

sans oublier la condition sur les opérateurs de transfert $(I_c^f)^T = c\, I_f^c$.
Ces inégalités nous assurent que la propriété d'approximation (4.10) est vérifiée.

Dans un cadre général, on ne peut pas aller plus loin dans la vérification des conditions d'utilisation. C'est pourquoi on s'intéresse, dans la suite, à un cas particulier. Cela nous permettra, non pas de donner des résultats généraux, mais d'illustrer le fonctionnement des méthodes multigrilles sur un cas assez simple pour maîtriser et comprendre son fonctionnement ; avant de passer à des cas plus complexes.

4.3 Application à une équation d'advection linéaire

4.3.1 Présentation de la configuration

Le modèle

Nous travaillons avec une simple équation d'advection à vitesse constante c, dans un domaine $\Omega = [0, L] \times [0, T]$ à une dimension d'espace. T est le temps d'intégration et L, la longueur du domaine.

$$\partial_t \mathbf{x} + c\, \partial_x \mathbf{x} = 0 \quad c > 0 \tag{4.13}$$

avec des conditions aux bords périodiques.

En pratique, ce n'est pas l'équation continue qui est résolue mais l'équation discrétisée. La méthode de discrétisation choisie est une méthode de différences finies.
On découpe le temps en N_t intervalles réguliers de durée Δt et l'espace en N_x intervalles réguliers de taille Δx.
Pour la discrétisation temporelle, on utilise un schéma décentré en temps. Quant à la discrétisation spatiale, on suppose qu'on a toujours une vitesse positive et que l'on peut donc utiliser un schéma d'Euler *upwind*, c'est-à-dire décentré et d'ordre 1.

On note $\mathbf{x}_n^j$, la valeur de $\mathbf{x}$ au point $\mathbf{x}(x = j\Delta x, t = n\Delta t)$.
En posant $\lambda = \dfrac{c\Delta t}{\Delta x}$ le nombre de Courant, l'équation discrète devient :

$$\mathbf{x}_{n+1}^j - \mathbf{x}_n^j = -\lambda \left(\mathbf{x}_n^j - \mathbf{x}_n^{j-1}\right) \tag{4.14}$$

On écrit le modèle sous une forme générale :

$$\mathbf{x}_{i+1} = M_{i,i+1}(\mathbf{x}_i)$$

où, $\mathbf{x}_i$ est la solution au temps i.

On en déduit l'expression de $\mathbf{x}_i$ en fonction de $\mathbf{x}_0$:

$$\mathbf{x}_i = M_{0,i}(\mathbf{x}_0) \tag{4.15}$$

où $M_{0,i} = M_{i-1,i} \circ M_{i-2,i-1} \circ \ldots \circ M_{0,1}$.

Les performances du schéma Euler upwind sont connues. Il est d'ordre un en temps et en espace et stable sous la condition $0 \leq \lambda \leq 1$. Le schéma numérique est hautement diffusif et l'équation continue modifiée n'est plus une équation d'advection pure, mais une équation d'advection-diffusion :

$$\partial_t \mathbf{x} + c\,\partial_x \mathbf{x} = \epsilon \partial_{xx} \mathbf{x} + O(\Delta x^2) + (\Delta t^2), \tag{4.16}$$

avec $\epsilon = \frac{c}{2}\,(\Delta x - c\,\Delta t)$, le coefficient de diffusion numérique.

Les observations

Dans le cadre d'expériences dites "jumelles", nous utilisons des observations artificielles, qui sont générées par le modèle à partir d'un état initial connu, que l'on note $\mathbf{x}_0^t$, en considérant que c'est l'état vrai.
Pour ce cas linéaire, on va supposer, qu'à chaque pas de temps et d'espace, correspond une observation. C'est-à-dire qu'on crée nos observations artificielles $\mathbf{y}$ en intégrant un modèle de même résolution que le modèle de la grille la plus fine. Comme nous utiliserons un facteur de raffinement entier, les observations seront représentées sur la grille grossière en utilisant une injection ; c'est à dire qu'on utilisera directement la valeur au point correspondant de la grille fine.
Pour simuler le bruit que l'on trouve sur les observations réelles, on ajoute au vecteur créé, une perturbation simulée par une gaussienne de moyenne nulle et de variance σ_o^2, notée $\mathcal{G}(0, \sigma_0^2)$.
Les observations utilisées sont donc :

$$\mathbf{y}_n = \mathbf{M}_{0,n}^f(\mathbf{x}_0^t) + \mathcal{G}(0, \sigma_0^2), \quad \forall n \in [0, N] \tag{4.17}$$

La fonction coût

On considère la fonction coût générale (4.1) dans le cas d'un modèle et d'un opérateur d'observation linéaire :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \tag{4.18}$$

Dans notre cas-test, les erreurs d'observation sont connues puisque les observations sont perturbées à l'aide d'un bruit gaussien $\mathcal{G}(0, \sigma_0^2)$. $\mathbf{R}_i$ est donc telle que, $\forall i \in [0, N-1]$:

$$\mathbf{R}_i^{-1} = \frac{1}{\sigma_o^2} \begin{pmatrix} 1 & 0 & & \\ 0 & 1 & \ddots & \\ & \ddots & \ddots & 0 \\ & & 0 & 1 \end{pmatrix} \tag{4.19}$$

La matrice de covariance d'erreur à l'ébauche est, quant à elle, définie de cette façon :

$$(\mathbf{B})^{-1} = \frac{\beta}{\sigma_b^2} \left(\mathbf{I} + l_1^2 (\nabla^h)^2 + l_2^4 (\triangle^h)^2 \right) \tag{4.20}$$

β est une constante positive utilisée pour donner plus ou moins d'importance au terme de régularisation. σ_b, l_1 et l_2 sont des coefficients qui peuvent avoir un sens statistique particulier. Typiquement, dans l'assimilation opérationnelle de données, $\mathbf{B}$ est représentative d'une distribution gaussienne de l'erreur. On peut alors approcher l'inverse de $\mathbf{B}$ par des dérivées spatiales d'ordre 1, avec ∇^h l'opérateur discret du gradient, et d'ordre 2 avec $\triangle^h$ l'opérateur discret du laplacien (voir [3]). Dans ce cas, σ_b^2 est la variance de l'erreur.
Pour avoir une représentation la plus proche possible d'une gaussienne, il faut $l_1^2 = \frac{\sigma_b^2}{2}$ et $l_2^4 = \frac{\sigma_b^4}{8}$. Cependant, nous approcherons une gaussienne en utilisant les coefficients :

$$l_1^2 = \frac{\sigma_b^2}{2}, \quad l_2^4 = \frac{\sigma_b^4}{16}$$

Ceci nous permettra d'exprimer facilement $\mathbf{B}^{\frac{1}{2}}$ pour effectuer un préconditionnement typique en assimilation de données (voir les travaux de Weaver [69]). En effet, dans ce cas, on peut écrire

$$\mathbf{B}^{-1} = \frac{\beta}{\sigma_b^2} \left(\mathbf{I} - \frac{\sigma_b^2}{4} \triangle^h \right)^2$$

La matrice de dérivées d'ordre 2, $\triangle^h$, sera définie en utilisant un schéma en différences finies centré.

Il faut aussi rappeler que les observations sont artificielles et la variable observée est directement $\mathbf{x}_i$. Comme $\mathbf{y}$ et $\mathbf{x}$ sont représentés sur la même grille de discrétisation, l'opérateur H est alors l'opérateur identité.
La fonction coût, ainsi définie, s'écrit :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2\sigma_o^2} \sum_{i=0}^{N-1} \|\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_i\|^2 + \frac{\beta}{2\sigma_b^2} \left\| \left(\mathbf{I} - \frac{\sigma_b^2}{4} \triangle^h \right) (\mathbf{x}_0 - \mathbf{x}^b) \right\|^2 \tag{4.21}$$

4.3.2 La propriété de lissage

Le lissage est conditionné par l'ellipticité du Hessien, comme indiqué section 3.2.1. Nous nous sommes placés dans le cas d'un modèle simple pour lequel il est relativement facile d'obtenir l'expression du Hessien dans l'espace de Fourier. Nous allons alors pouvoir étudier l'influence des divers paramètres qui entrent en jeu. Nous allons voir que le terme de régularisation ajoute naturellement de l'ellipticité au Hessien. Nous étudierons également l'influence du préconditionnement sur l'ellipticité.

Expression du Hessien

Le Hessien de notre système doit être elliptique. Son expression est similaire à la définition générale (4.9)

$$\mathcal{H} = \sum_i \left(\mathbf{H}_{t_i}\mathbf{M}_{0,i}\right)^T \mathbf{R}_i{}^{-1}\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1}$$

Nous cherchons ici son symbole pour étudier son comportement fréquentiel et en vérifier l'ellipticité. Nous envisageons le cas d'opérateurs continus puis celui des opérateurs discrétisés. Étudier le cas continu permet de s'abstraire des erreurs modèles et d'avoir une expression plus simple du gradient et du Hessien, en utilisant la solution connue du modèle. La fonction coût continue s'exprime de cette façon :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2\sigma_o^2}\int_0^T \|\mathbf{M}_{0,t}\mathbf{x}_0 - \mathbf{y}\|^2\, dt + \frac{\beta}{2\sigma_b^2}\left\|\left(\mathbf{I} - \frac{\sigma_b^2}{4}\triangle\right)(\mathbf{x}_0 - \mathbf{x}^b)\right\|^2$$

et le Hessien associé devient :

$$\mathcal{H}\delta\mathbf{x}_0 = \frac{1}{\sigma_o^2}\int_0^T \mathbf{M}_{0,i}^T\mathbf{M}_{0,i}\delta\mathbf{x}_0\, dt + \frac{\beta}{\sigma_b^2}\left(\mathbf{I} - \frac{\sigma_b^2}{4}\triangle\right)^2 \delta\mathbf{x}_0 \tag{4.22}$$

Comme $\mathbf{M}_{0,i}^T\mathbf{M}_{0,i}\delta\mathbf{x}_0 = \delta\mathbf{x}_0$ on obtient le symbole du Hessien en appliquant $\mathcal{H}$, dans l'espace de Fourier, à $\widetilde{\delta\mathbf{x}_0} = e^{imx}$, où m est le nombre d'onde :

$$\widetilde{\mathcal{H}} = \frac{T}{\sigma_o^2}\left[1 + \gamma\left(1 + \frac{\sigma_b^2}{2}m^2 + \frac{\sigma_b^4}{16}m^4\right)\right], \quad \text{avec}\, \gamma = \frac{\beta\sigma_o^2}{T\sigma_b^2}. \tag{4.23}$$

Dans le cas d'opérateurs discrétisés, nous utilisons le fait que le modèle linéaire est auto-adjoint, le symbole de Fourier du Hessien s'écrit :

$$\widetilde{\mathcal{H}} = \frac{\Delta t}{\sigma_o^2}\sum_i \widetilde{\mathbf{M}_{0,i}}^T\widetilde{\mathbf{M}_{0,i}} + \widetilde{\mathbf{B}^{-1}} = \frac{\Delta t}{\sigma_o^2}\sum_i \left|\widetilde{\mathbf{M}_{0,i}}\right|^2 + \widetilde{\mathbf{B}^{-1}} \tag{4.24}$$

Nous sommes capables de calculer le symbole du modèle $\mathbf{M}_{0,i}$ en exprimant l'équation (4.14) dans l'espace de Fourier :

$$\tilde{\mathbf{x}}_{i+1}^j = \underbrace{\left[1 - \lambda\left(1 - e^{-im\Delta x}\right)\right]}_{\widetilde{\mathbf{M}}_{i,i+1}(m)}\tilde{\mathbf{x}}_i^j$$

avec $\tilde{\mathbf{x}}_0^j = e^{im\,j\Delta x}$.
Le symbole du modèle est donc :

$$\begin{aligned} \widetilde{\mathbf{M}}_{0,i}(m) &= \widetilde{\mathbf{M}}^i(m) = \left[1 - \lambda\left(1 - e^{-im\Delta x}\right)\right]^i \\ |\widetilde{\mathbf{M}_{0,i}}|^2 &= \left[1 - 4\lambda(1-\lambda)\sin^2(m\Delta x/2)\right]^i \end{aligned} \tag{4.25}$$

Quant à $\widetilde{\mathbf{B}^{-1}}$, elle est définie en utilisant un schéma, en différences finies, centré pour les dérivées d'ordre 2 :

$$\widetilde{\mathbf{B}^{-1}} = \frac{\beta}{\sigma_b^2}\left[1 + \frac{4}{\Delta x^2}\frac{\sigma_b^2}{2}\sin^2\left(\frac{m\Delta x}{2}\right) + \frac{16}{\Delta x^4}\frac{\sigma_b^4}{16}\sin^4\left(\frac{m\Delta x}{2}\right)\right] \tag{4.26}$$

Etude de l'ellipticité

Nous différencions le cas continu du cas discret.
Dans le premier cas, le symbole du Hessien est donné par l'équation (4.23) :

$$\widetilde{\mathcal{H}}(m) = \frac{T}{\sigma_o^2}\left[1+\gamma\left(1+\frac{\sigma_b^2}{2}m^2+\frac{\sigma_b^4}{16}m^4\right)\right], \quad \gamma = \frac{\beta\sigma_o^2}{T\sigma_b^2} \tag{4.27}$$

Avec $\gamma = 0$, c'est-à-dire si la fonction coût n'inclut pas de terme de régularisation, alors le Hessien ne dépend pas du mode m. La méthode d'optimisation risque d'avoir un comportement similaire quelques soient les fréquences composant les erreurs successives de chaque itéré. L'utilisation de méthodes multigrilles est de moindre intérêt.
Cependant, en pratique, le terme de régularisation n'est jamais omis. Cela permet d'avoir un problème bien posé si la taille du vecteur d'observation est plus petite que celle du vecteur à estimer. Ce terme permet également de réduire l'impact du bruit des observations sur l'état estimé. Le paramètre γ est donc strictement positif. Dans ce cas, le Hessien devient un opérateur elliptique (voir (3.15)) car il existe $C > 0$, tel que

$$|\widetilde{\mathcal{H}}(m)| \geq C \sum_{p\in[0,2]} |m|^{2p},$$

et une méthode de relaxation adéquate réduira plus rapidement les hautes fréquences de l'erreur.

Cependant, le modèle utilisé n'est pas continu mais discret, car dans la plupart des cas la solution continue est inconnue. Le schéma discret que l'on utilise ici est donné par (4.14). L'approximation des dérivées partielles crée de la diffusion que l'on appelle diffusion numérique. Ainsi, l'équation réellement résolue n'est plus une équation d'advection pure mais une équation d'advection-diffusion (voir équation 4.16) avec un coefficient de diffusivité noté ϵ. Nous allons calculer le symbole du Hessien continu en utilisant l'équation modifiée.
L'expression $\int_0^T \mathbf{M}_{0,i}^T \mathbf{M}_{0,i}\delta\mathbf{x}_0$ de cette nouvelle équation est donc légèrement différente et le Hessien (4.27) prend maintenant en compte l'amplification de la diffusion du modèle direct et du modèle adjoint :

$$\widetilde{\mathcal{H}}(m) = \frac{1-e^{-2\epsilon m^2 T}}{\sigma_o^2 2\epsilon m^2} + \frac{\beta}{\sigma_b^2}\left(1+\frac{\sigma_b^2}{2}m^2+\frac{\sigma_b^4}{16}m^4\right) \tag{4.28}$$

Écrire la série de Taylor permet d'approcher le comportement du Hessien pour les basses fréquences. La série du Hessien continu donne à l'ordre 2 :

$$\widetilde{\mathcal{H}}(m) = \frac{T}{\sigma_o^2}\left(1-\epsilon T m^2+\gamma\left(1+\frac{\sigma_b^2}{2}m^2\right)\right) \tag{4.29}$$

Pour les grandes échelles ($m \ll 1$), ce symbole correspond à celui du Hessien discret (4.24), comme nous le vérifions maintenant :

$$\widetilde{\mathcal{H}} = \frac{\Delta t}{\sigma_o^2}\sum_i \left|\widetilde{\mathbf{M}_{0,i}}\right|^2 + \widetilde{\mathbf{B}^{-1}} \tag{4.30}$$

avec, dans le cas discret,

$$\widetilde{\mathbf{B}^{-1}} = \frac{\beta}{\sigma_b^2}\left[1+\frac{2\sigma_b^2}{\Delta x^2}\sin^2\left(\frac{m\Delta x}{2}\right)+\frac{\sigma_b^4}{\Delta x^4}\sin^4\left(\frac{m\Delta x}{2}\right)\right]$$

et

$$\left|\widetilde{\mathbf{M}_{0,i}}\right|^2(m) = \left[1 - 4\lambda(1-\lambda)\sin^2(m\Delta x/2)\right]^i$$

La série de Taylor à l'ordre 2 de ce symbole discrétisé nous donne :

$$\widetilde{\mathcal{H}}(m\Delta x) = \frac{N\Delta t}{\sigma_o^2}\left(1 - \frac{(N-1)\Delta x^2\lambda(1-\lambda)}{2}m^2\right) + \frac{\beta}{\sigma_b^2}\left(1 + \frac{\sigma_b^2}{2}m^2\right)$$

On sait que $\epsilon = \frac{c}{2}(\Delta x - c\Delta t)$ et $\lambda = \frac{c\Delta x}{\Delta t}$. Alors, en remplaçant l'expression de c par λ, ϵ est équivalent à $\frac{\Delta x^2}{2\Delta t}\lambda(1-\lambda)$.
En utilisant de plus, le fait que $N\Delta t = T$ et que $(N-1)\Delta t$ approche T, on obtient la simplification suivante :

$$\widetilde{\mathcal{H}}(m\Delta x) = \frac{T}{\sigma_o^2}\left(1 - \epsilon T m^2 + \gamma\left(1 + \frac{\sigma_b^2}{2}m^2\right)\right) \tag{4.31}$$

On retrouve bien la même expression que (4.29).

Dans le cas particulier où $\lambda = 1$ et $\epsilon = 0$, le modèle numérique ne diffuse pas et reproduit exactement la solution continue. L'équation résolue est donc bien une équation d'advection pure ($\epsilon = 0$) et le Hessien correspondant est elliptique (voir (4.27)).
Si le modèle n'est pas parfait ($\lambda < 1$ soit $\epsilon > 0$) alors il apparaît un terme d'ordre 2 dans le développement de Taylor. Mais ce terme a un coefficient négatif. Les fréquences, au lieu d'être bien réduites durant la résolution, vont être augmentées. Le terme de régularisation, celui qui a pour coefficient β, ajoute un terme d'ordre 2 avec un coefficient positif, il peut contrebalancer l'effet de la diffusion numérique (ou la diffusion du modèle). Nous reprenons le développement de Taylor du Hessien discret (4.31) :

$$\widetilde{\mathcal{H}}(m) = \frac{T}{\sigma_o^2}\left(1 - \epsilon T m^2 + \gamma\left(1 + \frac{\sigma_b^2}{2}m^2\right)\right)$$

avec $\gamma = \frac{\beta\sigma_o^2}{T\sigma_b^2}$, et $\epsilon = \frac{c}{2}(\Delta x - c\,\Delta t)$

Nous notons γ_{lim}, le coefficient γ qui annule le terme d'ordre 2 :

$$\gamma_{\text{lim}} = \frac{2\epsilon T}{\sigma_b^2} \tag{4.32}$$

Avec $\gamma > \gamma_{\text{lim}}$, le Hessien devient elliptique.

Nous avons étudié brièvement le comportement du Hessien pour les basses fréquences (m proche de 0). Pour compléter l'étude, nous allons calculer le Hessien (4.27) selon les différents modes m. On utilise pour cela un code numérique, pour lequel on spécifie des valeurs pour les différents paramètres. On fixe $T = 1$, $L = 4$, $c = 4$, $\sigma_o = 0.02$, $\sigma_b = 0.2$ et $\Delta x = 0.1$. La valeur de ϵ est égale à $\frac{c\Delta x}{2}(1-\lambda)$ et $\gamma_{\text{lim}} = 10((1-\lambda))$, d'après l'équation (4.32). À une constante multiplicative $(\frac{T}{\sigma_o^2})$ près, le Hessien dépend uniquement de ϵ et γ (voir expression (4.31)). L'étude se fait sur une grille unique. À Δx fixé, faire varier ϵ revient à faire varier λ. Nous étudions donc

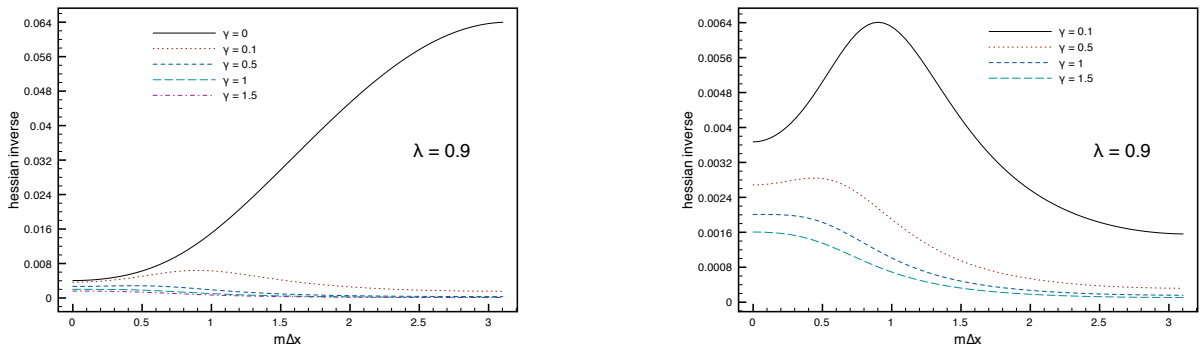

FIG. 4.2 – L'inverse du Hessien selon $m\Delta x$ pour différentes valeurs de γ avec $\lambda = 0.9$. À droite, lorsqu'on omet $\gamma = 0$.

l'influence des 2 paramètres : le nombre de Courant λ et le poids sur l'ébauche γ.

La figure 4.2 montre l'influence de γ sur le module de l'inverse du Hessien pour $\lambda = 0.9$.
L'ellipticité du Hessien se lit dans la forme de la courbe. Un opérateur elliptique réagit à une composante haute fréquence par un changement à haute fréquence local. Si le Hessien est elliptique, l'inverse de son symbole va donc réagir à une composante haute fréquence par un changement non local ; c'est-à-dire qu'il va réduire la fréquence. Donc si l'inverse du Hessien a une valeur plus grande pour les basses que pour les hautes fréquences, alors le Hessien est elliptique.
Quand le terme de régularisation est omis ($\gamma = 0$), la diffusion numérique rend le modèle moins contrôlable et le Hessien non elliptique. Avec $\gamma > 0$, le Hessien gagne en ellipticité.
Nous nous intéressons tout d'abord aux cas où $0 < \gamma < \gamma_{\text{lim}} = 10(1 - \lambda) = 1$. Pour les grandes échelles, le terme de régularisation n'est pas prédominant. Pour les plus petites échelles, il y a, au contraire, une compétition entre la diffusion artificielle induite par la discrétisation et le terme de régularisation. Cette compétition est mise en évidence par la présence d'un maximum local, le terme de régularisation ne devenant prédominant qu'à partir d'une certaine fréquence.
Lorsque $\gamma \geq \gamma_{\text{lim}} = 1$, le hessien devient complètement elliptique, le maximum local disparaît.

La figure 4.3 montre l'influence de λ sur le module de l'inverse du Hessien pour $\gamma = 1$.

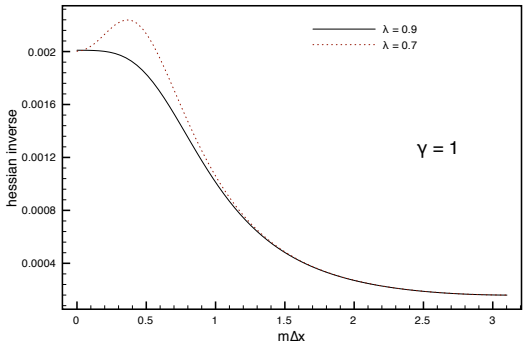

FIG. 4.3 – L'inverse du Hessien selon $m\Delta x$ pour différentes valeurs de λ avec $\gamma = 1$.

On observe qu'ajouter de la diffusion au modèle, en baissant le nombre de Courant, augmente le maximum local. Ceci n'est pas surprenant. En effet, le coefficient γ_{lim}, qui permet de contrebalancer l'effet de la diffusion, dépend de λ. Plus λ est petit, plus γ_{lim} est grand.

Le terme de régularisation est un terme nécessaire pour des équations diffusives et/ou des modèles numériques diffusifs. Dans le contexte de l'optimisation, il permet effectivement de rendre le système contrôlable. Même si les méthodes multigrilles ne sont pas utilisées, le terme de régularisation rendra l'optimisation sur une seule grille plus performante.
Mais il est aussi utile dans le cadre de méthodes multigrilles. Il permet, alors, à la méthode d'optimisation d'être en mesure de résoudre les hautes fréquences plus efficacement que les basses.

Effet du préconditionnement

Il s'agit d'effectuer un changement de variable qui fera en sorte que le Hessien sera mieux conditionné, ce qui veut dire qu'il aura des valeurs propres homogènes. Le conditionnement du Hessien permet d'améliorer la convergence d'une méthode d'optimisation. Mais quel est son influence sur l'ellipticité ? C'est ce que nous allons étudier dans ce paragraphe.

Nous appliquons le préconditionnement, détaillé section 2.3, en effectuant le changement de variable de $\mathbf{x}_0$ en $\bar{\mathbf{x}}_0 = \mathbf{B}^{-\frac{1}{2}}(\mathbf{x}_0 - \mathbf{x}^b)$. Ce préconditionnement est couramment utilisé. Son intérêt réside dans le fait qu'il assure un conditionnement fini, en imposant aux valeurs propres d'être au moins égales à 1.
Nous allons étudier les effets de ce changement sur l'expression du symbole du Hessien. Nous rappelons l'expression du Hessien sans préconditionnement :

$$\mathcal{H} = \sum_i (\mathbf{M}_{0,i})^T (\mathbf{R}_i)^{-1} \mathbf{M}_{0,i} + (\mathbf{B})^{-1}$$

Avec préconditionnement, le Hessien devient (voir équation (2.20))

$$\mathcal{H}_{\bar{\mathbf{x}}_0} = \mathbf{B}^{\frac{T}{2}} \left(\sum_i (\mathbf{M}_{0,i})^T (\mathbf{R}_i)^{-1} \mathbf{M}_{0,i} \right) \mathbf{B}^{\frac{1}{2}} + \mathbf{I} \tag{4.33}$$

Si nous considérons le symbole du Hessien, avec la valeur de $\widetilde{\mathbf{R}_i}$ précédente, on peut simplifier en utilisant l'égalité $\widetilde{\mathbf{B}^{\frac{T}{2}}\mathbf{B}^{\frac{1}{2}}} = \widetilde{\mathbf{B}}$:

$$\widetilde{\mathcal{H}}_{\bar{\mathbf{x}}_0}(m) = 1 + \frac{\Delta t}{\sigma_o^2} \widetilde{\mathbf{B}} \left(\sum_i |\widetilde{\mathbf{M}_{0,i}}(m)|^2 \right) \tag{4.34}$$

Dans le cas continu (modèle parfait) en remplaçant l'expression de $\widetilde{\mathbf{B}}^{-1}$ par l'expression trouvée équation (4.27) : $\widetilde{\mathbf{B}}^{-1} = \dfrac{\beta}{\sigma_b^2} \left(1 + \dfrac{\sigma_b^2}{2} m^2 + \dfrac{\sigma_b^4}{16} m^4 \right)$, nous obtenons :

$$\widetilde{\mathcal{H}}_{\bar{\mathbf{x}}_0}(m) = 1 + \frac{1}{\gamma \left(1 + \frac{\sigma_b^2}{2} m^2 + \frac{\sigma_b^4}{16} m^4 \right)} \tag{4.35}$$

ce qui donne, après développement de Taylor, à l'ordre 4 :

$$\widetilde{\mathcal{H}}_{\bar{\mathbf{x}}_0}(m) \approx 1 + \frac{1}{\gamma} \left(1 - \frac{\sigma_b^2}{2} m^2 - \frac{\sigma_b^4}{16} m^4 + \frac{\sigma_b^4}{4} m^4 \right) \tag{4.36}$$

Nous nous apercevons que le comportement du système conditionné est non elliptique car le terme d'ordre 2 a un coefficient négatif.
Dans le cas discret, nous vérifions la non-ellipticité induite par le préconditionnement Figure 4.4, où le symbole de l'inverse du Hessien a été calculé pour tous les nombres d'ondes m avec et sans préconditionnement pour $\gamma = 1$ et $\lambda = 0.9$. Nous calculons également le conditionnement du Hessien κ tel que,

$$\kappa(\mathcal{H}) = \frac{\max(\widetilde{\mathcal{H}})}{\min(\widetilde{\mathcal{H}})} \quad ,$$

avec et sans préconditionnement :

$$\kappa(\mathcal{H}) = 12.586, \quad \kappa(\mathcal{H}_{\bar{\mathbf{x}}_0}) = 1.985.$$

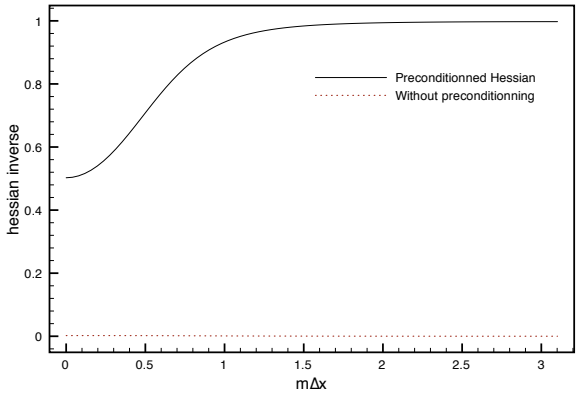

FIG. 4.4 – L'inverse du Hessien selon $m\Delta x$ pour $\gamma = \gamma_{\text{lim}} = 1$, $\lambda = 0.9$, avec et sans préconditionnement.

Ces résultats ne sont pas surprenants. En effet, le préconditionnement est utilisé pour rendre les courbures plus homogènes peu importe la direction. Si l'on étudie une courbe à une dimension, le préconditionnement modifie la courbure pour qu'elle soit plus proche d'un cercle. Il réduit alors la constante d'ellipticité, rend "moins" elliptique la courbe.
D'une manière générale, en réduisant l'ellipticité du Hessien, le préconditionnement réduit l'intérêt d'une méthode multigrille. En effet il améliore considérablement la résolution sur une seule grille, et l'effet d'une correction par une grille grossière est restreint.

Pourtant, le préconditionnement n'est pas toujours efficace. En effet, nous faisons le même calcul que précédemment avec des paramètres différents, dans le cas discret uniquement.
Tout d'abord, nous nous intéressons à la valeur γ_{lim} (voir équation (4.32)) qui rend elliptique le Hessien. Pour diminuer γ_{lim}, si l'on peut difficilement modifier ϵ et T, on peut augmenter σ_b. Ce que nous faisons en modifiant sa valeur de 0.2 à 0.4. γ_{lim} est alors égal à 0.25 et il faut $\gamma \geq 0.25$, pour un comportement elliptique.

Nous utilisons cependant un γ plus petit, $\gamma = 0.025$ et étudions l'ellipticité du nouveau Hessien ainsi que du Hessien préconditionné sur la figure 4.5. Les nouveaux conditionnements des

Hessiens deviennent alors :

$$\kappa(\mathcal{H}) = 11.0437, \quad \kappa(\mathcal{H}_{\bar{\mathbf{x}}_0}) = 40.251 \tag{4.37}$$

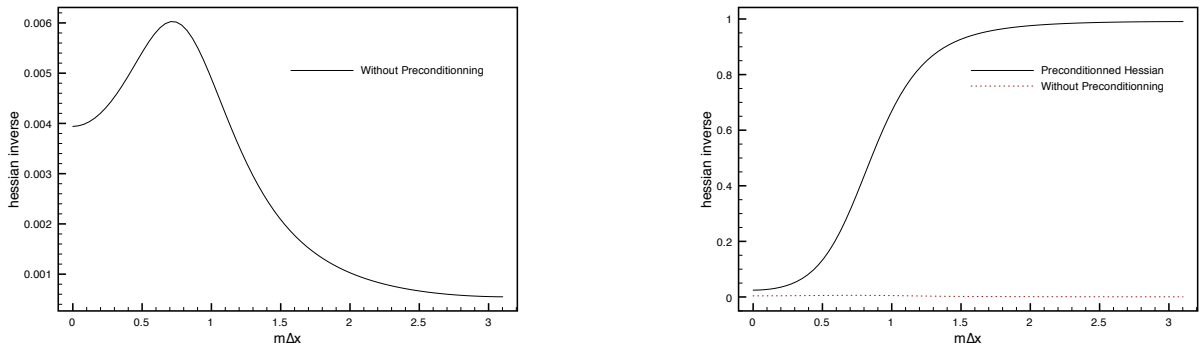

FIG. 4.5 – L'inverse du Hessien selon $m\Delta x$ pour $\gamma = 0.025 < \gamma_{\text{lim}} = 0.25$, $\lambda = 0.9$, avec et sans préconditionnement, à droite ; et sans préconditionnement uniquement à gauche.

Nous constatons que le Hessien non préconditionné n'est évidemment pas complètement elliptique mais également que le préconditionnement n'est pas adéquat étant donné qu'il augmente la valeur du conditionnement ; de plus il rend moins elliptique, fréquentiellement, le système. L'augmentation du conditionnement est due au fait que le symbole du Hessien préconditionné (4.35) tend vers $1 + 1/\gamma$ (qui est égal à 41 dans l'exemple ci-dessus), pour des basses fréquences et vers 1 pour des hautes fréquences. De petites valeurs de γ peuvent donner un grand écart entre la plus grande et la plus petite des valeurs propres.
Nous notons donc que γ est un paramètre important pour le préconditionnement, il représente le rapport entre les variances des erreur à l'ébauche et des erreurs aux observations. Nous pourrions calculer le conditionnement des Hessiens en fonction des différents paramètres, sans préconditionnement. Cela compléterait l'étude, et nous pourrions savoir quel paramètre est le plus sensible, et pour quelle valeur de γ, le conditionnement devient plus grand, après préconditionnement par $\mathbf{B}$. Cependant, nous ne sommes pas entrés dans ces détails.

Le préconditionnement par $\mathbf{B}$ n'est donc pas toujours intéressant. Les méthodes multigrilles, inversement, semblent pouvoir être applicables efficacement sur des classes plus larges de fonctions coût. En effet, l'ellipticité est plus facilement contrôlable que le conditionnement, dans le sens où il est plus facile et moins coûteux d'ajouter de l'ellipticité à un système que de trouver un bon conditionnement.

4.3.3 Étude de la convergence

Le passage par la grille grossière

Parce que le cas étudié est déjà d'une résolution peu élevée, on ne va travailler que sur deux grilles. Dans ce paragraphe, les symboles calculés sont tous discrets. Le comportement du passage par la grille grossière peut être étudié en utilisant l'expression du facteur de convergence (4.6) dans le cas de notre système optimal :

$$\mathbf{K} = I - I_c^f (\mathcal{H}^c)^{-1} I_f^c \mathcal{H}^f \tag{4.38}$$

Ce facteur de convergence est calculé **en supposant que l'on a trouvé la solution optimale sur la grille à basse résolution**.
Pour que les méthodes multigrilles soient efficaces, le facteur de convergence doit être proche de 0 pour les grandes échelles. En effet, les petites échelles sont supposées être bien résolues par une minimisation sur la grille fine.

La grille à haute résolution a une taille de discrétisation de $\Delta x^f = 0.1$, c'est-à-dire qu'il y a $N_x^f = 40$ points en espace. (On rappelle que la longueur du domaine est de 4.) La grille à basse résolution comporte deux fois moins de points. Cela correspond à $\Delta x^c = 0.2$ et $N_x^c = 20$. On pourrait ne raffiner qu'en espace, mais pour que le modèle de la grille grossière reste relativement proche de celui de la grille fine, il faut utiliser également un raffinement du même ordre en temps. En effet, raffiner en espace modifie la valeur du nombre de Courant λ du modèle numérique (4.14). Les pas de temps utilisés sont donc $\Delta t^f = \dfrac{\lambda \Delta x^f}{c}$ et $\Delta t^c = 2\Delta t^f$, avec $c = 4$ et λ variable.
Ce type de raffinement nous donne le même nombre de Courant sur les deux grilles, mais la diffusion numérique ϵ est deux fois plus grande sur la grille grossière.

Nous choisissons la matrice de covariance d'ébauche définie pour ses caractéristiques statistiques, c'est-à-dire que la fonction coût utilisée est la fonction (4.21) avec $l_1^2 = \frac{\sigma_b^2}{2}$ et $l_2^4 = \frac{\sigma_b^4}{16}$. Le symbole du Hessien continu correspondant est celui décrit équation (4.27). Cette matrice de covariance d'ébauche n'est pas exactement une gaussienne, comme nous l'avons déjà précisé section 4.3.1 ; mais elle va nous permettre d'exprimer $\mathbf{B}^{1/2}$ et d'étudier la minimisation de la fonction coût préconditionnée par $\mathbf{B}$.

Pour pouvoir calculer le Hessien, il nous manque alors la définition de quelques paramètres présents dans la fonction coût : T, σ_o, σ_b et γ. Ces paramètres peuvent être différents sur les deux grilles sauf T que l'on a pris égal à 1. Il est tout à fait intéressant de pouvoir faire varier ces paramètres selon la taille de la grille. En effet, σ_o contient à la fois les erreurs d'observation et les erreurs de représentativité. La confiance que l'on a dans les observations peut varier selon la taille de discrétisation du modèle. Plus le modèle est à basse résolution, moins bien l'on va résoudre les hautes fréquences des observations.
On suppose néanmoins que les trois derniers paramètres ne dépendent pas de la discrétisation. Cette hypothèse est évidemment réfutable.
Pour le modèle basse résolution, les observations sont également disponibles en chaque pas de temps et d'espace. C'est-à-dire que 4 fois moins d'observations sont disponibles. On impose pour cette étude $\sigma_o = 0.02$, et $\sigma_b = 0.2$ sur les deux grilles. On se réduit à étudier l'influence de la diffusivité du modèle numérique λ et l'influence du poids mis sur le terme de régularisation γ.
À l'aide de tous ces paramètres, nous calculons le Hessien que ce soit sur la grille fine ou la grille grossière en utilisant l'expression discrète (4.30).

Pour exprimer le facteur de convergence de l'expression (4.38), il ne manque plus que la définition des opérateurs de transfert.
Pour la restriction, on choisit l'opérateur I_f^c nommé "full weighting" et pour l'interpolation I_c^f,

l'interpolation linéaire. On rappelle que ces opérateurs sont définis comme tels :

$$I_f^c = \frac{1}{2}\left[\begin{array}{ccc} \frac{1}{2} & 1 & \frac{1}{2} \end{array}\right]_f \quad I_c^f = \left[\begin{array}{c} \frac{1}{2} \\ 1 \\ \frac{1}{2} \end{array}\right]_c \tag{4.39}$$

et qu'ils ont le même symbole, sur une grille de taille Δx :

$$\widetilde{I_f^c}(m) = \widetilde{I_c^f}(m) = 1 - \sin^2(\frac{m\Delta x}{2}) \tag{4.40}$$

Ils doivent être d'ordres suffisamment élevés pour vérifier la relation (3.20). Le modèle parfait donne un système d'ordre 1. Il est de 2 lorsque le schéma numérique induit de la diffusion. Les opérateurs choisis sont donc adaptés, car étant d'ordre 2.

L'étude du facteur de convergence $\mathbf{K}$ selon différentes fréquences ou valeurs de $m\Delta x^f$, permet de s'assurer de la validité du choix des différents opérateurs et paramètres. Figure 4.6, est représenté le symbole de $\mathbf{K}$ pour différentes valeurs de γ à gauche et différentes valeurs de λ à droite.

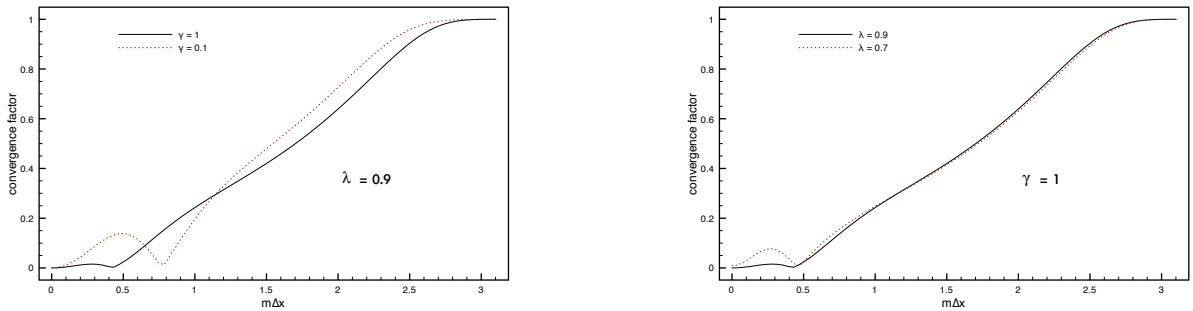

FIG. 4.6 – Le facteur de convergence d'un passage par la grille grossière selon $m\Delta x$ pour différentes valeurs de γ ($\lambda = 0.9$) à gauche et différentes valeurs de λ à droite ($\gamma = 1$).

Le maximum local que l'on trouve pour les basses fréquences entre $[0.2; 0.8]$ est le même que celui observé sur les courbes de l'inverse du Hessien, figure 4.2. Nous rappelons qu'il est dû au fait que le terme de régularisation, qui rend le comportement elliptique, n'est pas effectif pour les très basses fréquences.
Les deux paramètres étudiés ont une influence légèrement différente sur le comportement des méthodes multigrilles. Augmenter γ permet de lisser la courbe. Faire varier λ permet au contraire d'agir plus efficacement sur la valeur de ce maximum, cependant sans modifier la forme générale de la courbe. Ceci dit, en pratique, si l'on peut difficilement influencer sur λ qui est un paramètre induit par les schémas numériques choisis et l'équation originelle, on fait plus facilement varier γ.
Dans tous les cas, les grandes échelles sont beaucoup mieux résolues que les petites et on peut espérer que numériquement la correction par grille grossière réduise efficacement les grandes échelles.

Nous étudions, Figure 4.7, le facteur de convergence associé à la fonction coût préconditionnée, c'est à dire en utilisant le symbole du Hessien (4.34), dans le cas discret c'est à dire avec un

modèle diffusif.

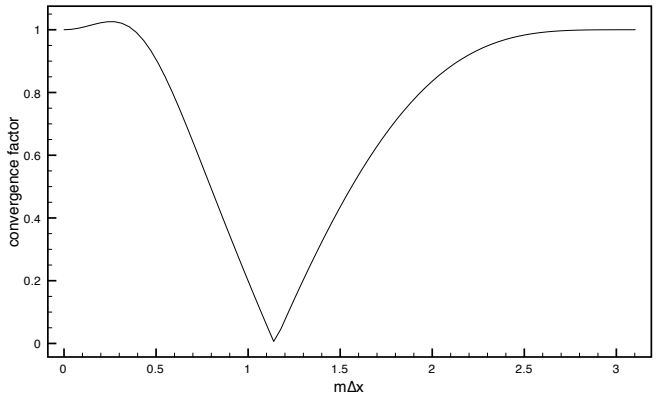

FIG. 4.7 – Le facteur de convergence d'un passage par la grille grossière selon $m\Delta x$, avec préconditionnement et pour $\lambda = 0.9$, $\gamma = 1$.

Il s'avère que les basses fréquences ne sont pas bien résolues par la correction grille grossière. Elles sont même amplifiées lorsque le facteur dépasse la valeur 1. Le préconditionnement semble ne pas être compatible avec les méthodes multigrilles. Il y a cependant une bande de fréquences pour laquelle le passage par la grille grossière est efficace.

Étude d'un V-cycle

Nous allons maintenant présenter les résultats numériques d'une résolution complète d'un cycle d'une méthode multigrille.
Nous faisons quelques itérations de la méthode d'optimisation sur la grille à haute résolution pour résoudre le système d'assimilation, pour ensuite passer l'état initial sur la grille basse résolution et résoudre l'équation résiduelle sur la grille basse résolution. La dernière étape consiste à refaire quelques itérations d'optimisation sur la grille haute résolution.

Il nous faut, avant de présenter les résultats, définir les derniers paramètres non utilisés jusque là. L'ébauche et les observations n'ont pas encore été choisis.
On va considérer que les observations de notre cas test sont créées comme décrit section 4.3.1 et équation (4.17), en bruitant

$$\mathbf{x}_0^t = \begin{cases} 1 & \text{pour } 1 \leq x \leq 2 \\ 0 & \text{sinon} \end{cases} \tag{4.41}$$

qui est l'état initial vrai.
On rappelle que les observations sont supposées disponibles en chaque pas de temps et d'espace que ce soit sur la grille fine ou la grille grossière.
La première approximation de l'état initial, ou l'ébauche $\mathbf{x}^b$, est prise égale à l'état initial vrai bruité avec 3 sinus de fréquences différentes de longueurs d'onde $4\Delta x$, $8\Delta x$ et $20\Delta x$. Ce calcul est fait sur la grille fine. L'ébauche de la grille grossière est égale à $(\mathbf{x}^b)^c = I_f^c(\mathbf{x}^b)^f$.
Les autres paramètres sont fixés aux mêmes valeurs que pour les tests précédents, le terme de régularisation est celui décrit par l'équation (4.21) avec $l_1^2 = \frac{\sigma_b^2}{2}$ et $l_2^4 = \frac{\sigma_b^4}{16}$ et l'on fixe λ à 0.9 et

γ égal à $\gamma_{\text{lim}} = 1$. Le hessien est complétement elliptique.

On peut calculer l'état optimal $\mathbf{x}_0^a$ en minimisant jusqu'à convergence sur la grille haute résolution. On l'a représenté figure 4.8, avec l'ébauche et l'état initial vrai. On observe que l'état optimal est différent de l'état vrai et très proche de l'ébauche. Ceci est dû au fait que le poids sur le terme à l'ébauche est trop grand, ou que l'ébauche est trop éloignée de l'état vrai. Cependant, nous avons voulu garder la valeur de γ_{lim}, dans un premier temps tout du moins, pour des raisons théoriques. D'autres valeurs de γ sont testées dans la suite.

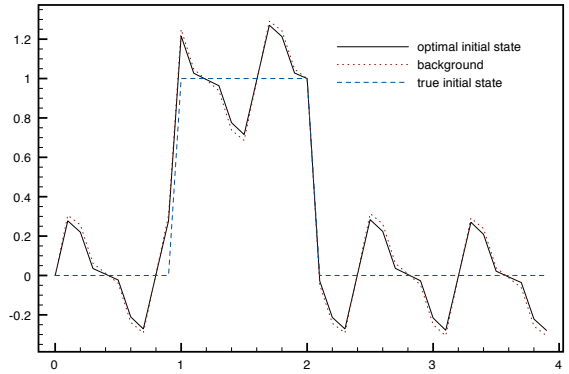

FIG. 4.8 – Ébauche, état initial vrai et optimal

On peut observer, Figure 4.9, l'erreur initiale représentée dans l'espace de Fourier. Elle est égale à l'ébauche moins l'état optimal. On retrouve les 3 fréquences que l'on a ajoutées à l'état initial vrai. En effet, on se trouve dans un cas linéaire et les fréquences ajoutées à l'ébauche sont prédominantes par rapport au bruit des observations. C'est donc les 3 modes ajoutés à l'ébauche qui apparaissent dans l'erreur. On va pouvoir étudier les différences d'impact sur chacune de ces fréquences tout au long de l'algorithme des méthodes multigrilles.

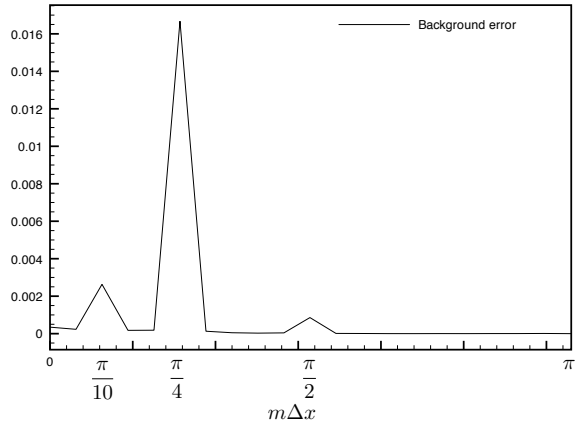

FIG. 4.9 – Erreur initiale représentée dans l'espace de Fourier

Le schéma 4.10 décrit l'algorithme multigrille utilisé, on part de l'ébauche pour faire 5 itérations d'optimisation du gradient conjugué sur la grille fine. On passe ensuite sur la grille grossière et on résout le système grossier en effectuant 10 itérations de la même méthode d'optimisation. La correction est alors transférée sur la grille fine. On effectue ensuite 5 dernières itérations sur la grille fine. 10 itérations sur la grille grossière suffisent pour la convergence de l'équation résiduelle. On se trouve ainsi dans les conditions théoriques d'analyse du passage par la grille grossière.

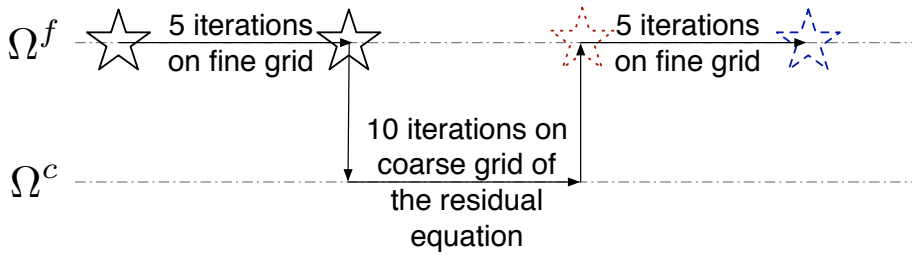

FIG. 4.10 – Schéma expliquant la méthode multigrille utilisée pour le cas test linéaire

On compare cette méthode avec la méthode qu'on appelle monogrille qui consiste à n'effectuer que 10 itérations de la méthode d'optimisation du gradient conjugué sur la grille fine.

Sur les figures 4.11 et 4.12, on représente, dans l'espace de Fourier, l'erreur entre l'état initial analysé et l'état initial optimal $\mathbf{x}_0^a$, pour chacune des étapes de la méthode multigrille. Les erreurs sont représentées sur deux figures différentes pour des raisons d'échelles. Sur la première figure, on observe les erreurs de l'ébauche et de l'état trouvé après les 5 premières itérations de la grille fine. Sur la deuxième figure, on trouve les erreurs après chaque étape du multigrille ainsi que l'erreur de la méthode monogrille.

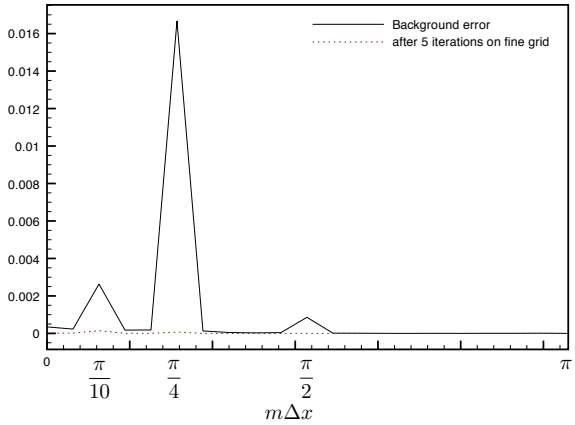

FIG. 4.11 – Erreur initiale et après 5 itérations sur la grille fine dans l'espace de Fourier

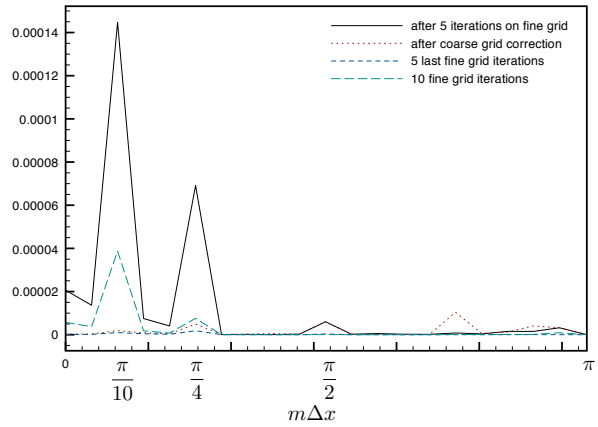

Fig. 4.12 – Erreur dans l'espace de Fourier après chaque étape de la méthode multigrille, comparaison avec la méthode monogrille.

On interprète, tout d'abord, les résultats de la méthode multigrille. Après les 5 premières itérations, l'erreur a, semble-t-il, été aussi bien réduite pour les hautes que pour les basses fréquences (Figure 4.12). Pourtant, cette courbe, à elle seule, ne permet pas de conclure sur le comportement fréquentiel de la méthode d'optimisation. En effet, on peut apercevoir l'amplification d'une haute fréquence, qui est surement due au fait que la méthode de gradient conjugué n'est pas une méthode de relaxation.
Les 10 itérations de la grille grossière ont bien éliminé les basses fréquences (Figure 4.12) mais font réapparaître quelques hautes fréquences. Ces hautes fréquences parasites sont des preuves supplémentaires que la méthode d'optimisation n'a pas tout à fait la propriété de lissage. Sur la grille grossière, elle a ajouté des hautes fréquences non présentes à l'origine. A moins que ces hautes fréquences proviennent des défauts de l'opérateur d'interpolation. Ces fréquences sont heureusement réduites par les 5 dernières itérations sur la grille fine. Le fonctionnement des méthodes multigrilles est donc satisfaisant.

Sur la figure 4.12 ; en comparant l'erreur finale de la méthode multigrille avec l'erreur donnée après 10 itérations sur la grille fine uniquement, nous constatons que la méthode multigrille est plus efficace, particulièrement pour les basses fréquences.

Préconditionnement par $\mathbf{B}^{1/2}$

La convergence de ce cas test simple est atteinte très rapidement. Il sera donc difficile de distinguer les effets du préconditionnement qui, rappelons-le, est utilisé pour rendre plus rapide la convergence.

Le préconditionnement consiste à effectuer le changement de variable (voir section 2.3) :

$$\overline{\mathbf{x}}_0 = \mathbf{B}^{-\frac{1}{2}}(\mathbf{x}_0 - \mathbf{x}^b)$$

En pratique, la fonction coût est exprimée en fonction de $\overline{\mathbf{x}}_0$, en partant de $\overline{\mathbf{x}}_0 = 0$ et le modèle est intégré après l'application du changement de variable suivant :

$$\mathbf{x}_0 = \mathbf{x}^b + \mathbf{B}^{\frac{1}{2}}\overline{\mathbf{x}}_0 \tag{4.42}$$

Pour cet exemple numérique, il nous faut non plus le symbole de $\mathbf{B}^{1/2}$ mais son expression dans l'espace réel. L'expression de $\mathbf{B}^{-1}$ (4.20), rappelée ci dessous :

$$\mathbf{B}^{-1} = \frac{\beta}{\sigma_b^2}\left(\mathbf{I} + l_1^2\nabla^2 + l_2^4\triangle^2\right)$$

a l'avantage de pouvoir se mettre sous la forme :

$$\mathbf{B}^{-1} = \frac{\beta}{\sigma_b^2}\left(\mathbf{I} - \frac{\sigma_b^2}{2}\triangle\right)^2$$

dans notre cas où $l_1^2 = \frac{\sigma_b^2}{2}, \quad l_2^4 = \frac{\sigma_b^4}{16}$. Cette formulation permet de calculer $\mathbf{B}^{-\frac{1}{2}}$ facilement :

$$\mathbf{B}^{-\frac{1}{2}} = \frac{\sqrt{\beta}}{\sigma_b}\left(\mathbf{I} - \frac{\sigma_b^2}{2}\triangle\right) \tag{4.43}$$

Le cas 1-D nous permet de mettre $\mathbf{B}^{-\frac{1}{2}}$ sous forme matricielle et d'appeler une procédure d'inversion pour effectuer (4.42).
Nous avons implémenté, ici, le même test que dans le cas non préconditionné pour $\lambda = 0.9$, $\gamma = \gamma_{\text{lim}} = 1$. Nous avons aussi testé le cas où $\gamma_{\text{lim}} = 0.25$ et $\gamma = 0.025$. Le permier test nous donne un Hessien complètement elliptique mais un état initial optimal très proche de l'ébauche, la deuxième valeur donne un état optimal un peu plus près de l'état vrai mais est aussi un cas où le préconditionnement n'est théoriquement pas satisfaisant. Le reste de la configuration est exactement le même que précédemment. Sur la Figure 4.13, nous comparons les deux configurations, avec à gauche l'état initial optimal et à droite l'erreur initiale dans l'espace de Fourier.

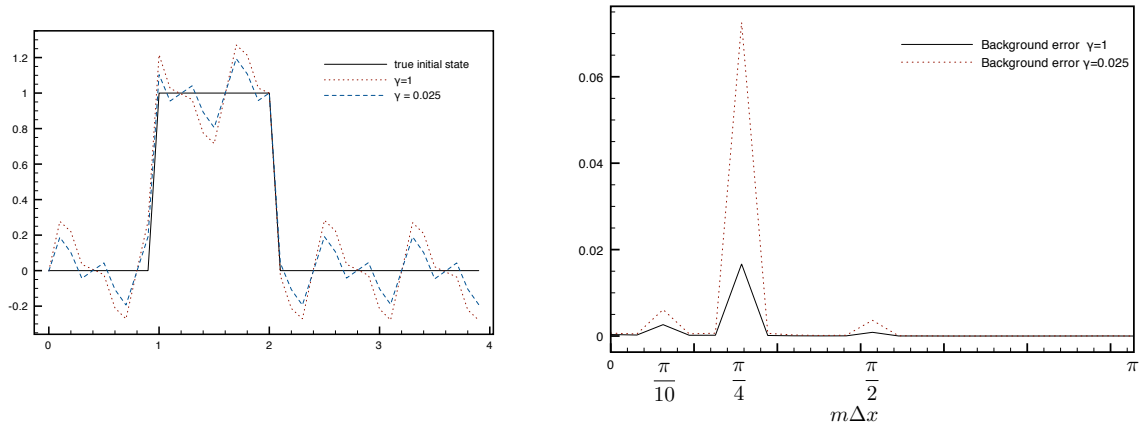

FIG. 4.13 – Comparaisons des deux configurations avec $\gamma = 1$ et $\gamma = 0.025$. A gauche, les états initiaux optimum et à droite l'erreur initiale en mode de Fourier.

Nous effectuons pour la fonction coût préconditionnée et ces deux valeurs de γ un cycle de

multigrille, composé de 5 itérations sur la grille fine, 10 itérations sur la grille grossière puis 5 dernières itérations sur la grille fine. Nous nommons cet algorithme "P Multigrid" sur les graphiques suivants. Nous comparons les résultats avec ceux obtenus après 10 itérations sur la grille fine, méthode que nous appelons "P Monogrid". Nous observons les erreurs dans l'espace de Fourier, pour le "P Monogrid" et le "P Multigrid", comparées aux résultats précédents, nommés "Monogrid" pour 10 itérations sur la grille fine et "Multigrid" pour un cycle de multigrilles. Le graphique 4.14 nous montre les erreurs obtenues pour $\gamma = \gamma_{\text{lim}}$, et le graphique 4.15, pour $\gamma = 0.025$.

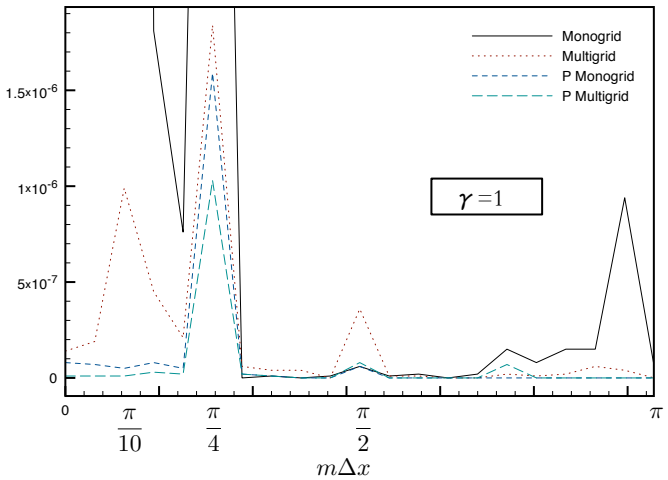

FIG. 4.14 – Comparaisons des algorithmes monogrilles et multigrilles avec ou sans préconditionnement pour $\gamma = \gamma_{\text{lim}} = 1$.

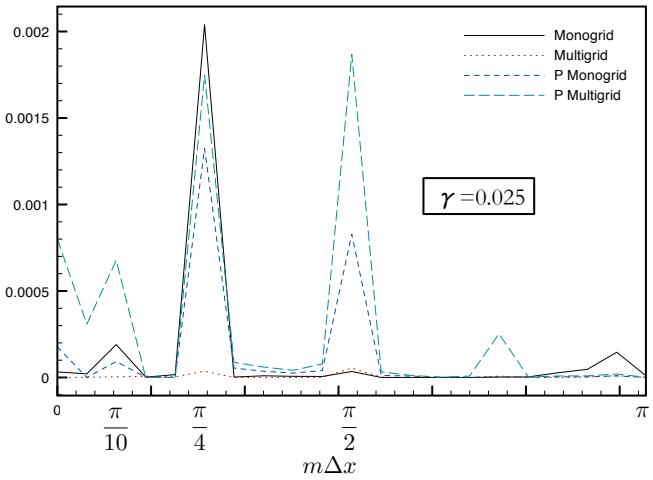

FIG. 4.15 – Comparaisons des algorithmes monogrilles et multigrilles avec ou sans préconditionnement pour $\gamma = 0.025 < \gamma_{\text{lim}} = 0.25$.

Dans le cas où $\gamma = 1$, le Hessien est tout à fait elliptique, il semblerait que le multigrille améliore le résultat avec et sans préconditionnement. Cependant les erreurs étant très faibles, il est difficile

d'en tirer une conclusion.
Il est à noter que le préconditionnement n'améliore pas toujours les résultats comme c'est le cas pour $\gamma = 0.025$. Dans ce cas, même si le Hessien n'est pas tout à fait elliptique, le multigrille demeure plus efficace que le monogrille, sans préconditionnement. Cependant le multigrille avec préconditionnement est, au contraire, moins efficace que le monogrille avec préconditionnement.

Il est intéressant de remarquer ici que diminuer la valeur de γ peut être nécessaire.
En effet, dans notre test, l'état optimal reste trop près de l'ébauche et loin de l'état vrai ; car le terme de régularisation a un poids trop important. Réduire ce poids est possible car l'ellipticité ajoutée par le terme à l'ébauche, et utilisée pour rendre le système mieux posé, peut également être apportée par la matrice $\mathbf{R}$ du terme aux observations. Il faudrait, pour cela, utiliser par exemple une matrice $\mathbf{R}^{-1}$ définie de la même façon que notre matrice $\mathbf{B}^{-1}$. Ceci faisant, le poids du terme à l'ébauche peut être diminué, et l'état optimal dépendrait alors beaucoup moins de l'ébauche.

Toutefois si diminuer γ a moins de conséquences sur l'ellipticité du système, le préconditionnement par $\mathbf{B}$ peut faire diverger le conditionnement, au-delà d'une certaine valeur de γ (voir l'exemple (4.37)).
Dans ces cas-là, il faut utiliser un autre préconditionnement, qui conditionnerait le terme aux observations. Mais ce type de préconditionnement est plus difficile à trouver car le terme aux observations fait apparaître le modèle et non pas seulement la matrice de covariance d'erreur $\mathbf{R}$.
Dans ces cas-là, les méthodes multigrilles peuvent être une alternative au préconditionnement.

4.4 Conclusion

Nous avons étudié dans ce chapitre les méthodes multigrilles appliquées à l'assimilation de données, dans le cas d'un modèle linéaire. Après avoir adapté les différentes conditions d'utilisation à un cas général, nous nous sommes concentrés sur un cas plus particulier d'un modèle d'advection à une dimension d'espace.
L'ellipticité du Hessien a pu être étudiée dans le détail et il s'est avéré que le terme de régularisation ne doit pas être omis. Nous avons pu également constaté l'influence du préconditionnement par $\mathbf{B}$ sur le comportement fréquentiel. Un bon préconditionnement réduit la constante d'ellipticité du système, en modifiant la fonction coût pour que ses courbes soient plus homogènes quelque soit la direction. Ceci faisant, il peut rendre les méthodes multigrilles, appliquées sur un système préconditionné, moins efficaces.
Nous nous sommes ensuite placés dans le cas où la forme du terme de régularisation donne un Hessien elliptique pour étudier l'effet d'un passage par la grille grossière. Les opérateurs de transfert choisis sont très simples, mais le facteur de convergence d'une correction grille grossière est acceptable. Cependant celui du système préconditionné est beaucoup plus grand pour les basses fréquences.
Enfin, l'étude sur un cycle complet d'une méthode multigrille, avec l'utilisation d'une méthode d'optimisation en gradient conjugué, prouve l'efficacité d'un passage par la grille grossière. Le bon comportement des méthodes multigrilles est donc vérifié, même si toutes les conditions de bon fonctionnement ne sont pas prouvées. Ainsi nous n'avons pas la certitude que la méthode d'optimisation ait la propriété de lissage. De plus, nous sommes ici dans un contexte particulier

où de nombreuses observations sont à notre disposition. Dans les applications pratiques, la taille du vecteur de contrôle est plus grande que celle du vecteur d'observation. Il est alors difficile de généraliser les bons résultats obtenus dans ce cas simple.
Nous avons testé également la minimisation de la fonction coût préconditionnée. Dans le cas où le préconditionnement par $\mathbf{B}$ améliore le conditionnement, l'utilisation des méthodes multigrilles n'a plus autant d'intérêt. Cependant utiliser les méthodes multigrilles sur un système préconditionné ne détériore pas la convergence. Les méthodes multigrilles peuvent donc être utilisées sur une large classe de fonctions coûts, alors que l'efficicacité du préconditionnement par $\mathbf{B}$ dépend de la définition de $\mathbf{R}$ et $\mathbf{B}$ et peut s'avérer être un mauvais préconditionnement. De plus, l'ellipticité est plus facilement contrôlable que le conditionnement.

Les résultats des méthodes multigrilles sur un modèle linéaire sont encourageants. Nous étudions l'extension aux cas non linéaires dans le chapitre suivant.

5

L'EXTRAPOLATION À LA NON LINÉARITÉ

Sommaire

5.1 Les algorithmes étudiés 108
5.1.1 La méthode FAS 109
5.1.2 La méthode Gauss Newton - multigrille (GN-MG) 110
5.1.3 La méthode multi-incrémentale 112
5.1.4 Les contraintes d'utilisation 114
5.2 Spécificité des tests non linéaires 115
5.2.1 Le modèle et les observations 115
5.2.2 La fonction coût 116
5.2.3 Les deux grilles 116
5.2.4 Paramètres et configuration 117
5.3 Les multigrilles éprouvées 118
5.3.1 Les variables d'initialisation et les diagnostics utilisés 118
5.3.2 Erreurs d'approximation 119
5.3.3 Paramètres critiques 122
5.3.4 Monogrille vs. Multirésolution 125
5.3.5 Préconditionnement 128
5.4 Conclusion 129

Notre but est d'adapter les méthodes multigrilles au problème d'assimilation de données non linéaire :

$$\text{Trouver } \mathbf{x}_0^a \text{ tel que } \min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0) = \mathcal{J}(\mathbf{x}_0^a)$$

La fonction coût est décrite, ici, sous la forme (voir équation (1.1)) :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[H_{t_i}(M_{0,i}(\mathbf{x}_0)) - \mathbf{y}_{t_i}]^T\mathbf{R}_i^{-1}[H_{t_i}(M_{0,i}(\mathbf{x}_0)) - \mathbf{y}_{t_i}] + \frac{1}{2}[\mathbf{x}_0 - \mathbf{x}^b]^T\mathbf{B}^{-1}[\mathbf{x}_0 - \mathbf{x}^b] \quad (5.1)$$

avec cette fois, un modèle et un opérateur d'observation non linéaires. En effet, le but de ce chapitre est de vérifier le comportement des méthodes multigrilles dans le cas de non-linéarités. Au chapitre précédent, nous avons étudié le comportement fréquentiel des méthodes multigrilles. Cette étude est insuffisante dans le cas d'un modèle non linéaire. On ne peut pas calculer le facteur de convergence des passages par la grille grossière car le modèle n'agit pas linéairement sur chaque fréquence.
Nous voulons vérifier que les méthodes multigrilles, utilisées dans ces conditions, peuvent également améliorer la convergence. Ce chapitre se concentre sur l'étude numérique de différents algorithmes adaptés aux systèmes non linéaires.
Nous implémentons deux algorithmes multigrilles. Le premier est une application directe du FAS sur l'assimilation de données du modèle non linéaire et le second est une méthode Gauss-Newton Multigrille, encore appelée GN-MG. Cette dernière méthode est l'algorithme multigrille appliqué à chaque minimisation induite par la méthode incrémentale (ou méthode Gauss-Newton).
Nous comparons ces méthodes avec les méthodes monogrilles correspondantes ainsi qu'avec la méthode multi-incrémentale.

Nous commençons par présenter les algorithmes multirésolutions utilisés ; le FAS, le GN-MG et le multi-incrémental. Ces algorithmes diffèrent par leurs propriétés et conditions de convergence qui sont plus ou moins sensibles aux non-linéarités.
Nous testons ces méthodes sur deux configurations d'une équation de Burgers 1-D et vérifions la validité de ces conditions de convergence sur chacun des tests, en étudiant les erreurs de linéarisation et d'approximation par la grille grossière obtenues.
Nous nous intéressons ensuite aux paramètres de configuration des méthodes multigrilles. Nous allons discuter de leur influence afin de les régler au mieux.
Enfin, après avoir fixé les différents paramètres, nous pouvons confronter toutes les méthodes en comparant les résultats. Nous implémentons alors le préconditionnement pour en étudier les effets.

5.1 Les algorithmes étudiés

Dans ce paragraphe, nous rappelons le fonctionnement des différents algorithmes que nous allons tester dans les cas non-linéaires.

Nous nous intéressons à deux algorithmes monogrilles. Le premier est tout simplement la minimisation de la fonction coût (5.1) sur une grille haute résolution. Le deuxième consiste à appliquer la méthode incrémentale pour résoudre cette fonction coût haute résolution. Ces algorithmes dépendent uniquement de la méthode de minimisation choisie. Nous ne nous attardons pas sur leur fonctionnement déjà décrit dans les deux premiers chapitres. De plus, chaque algorithme, monorésolution et multirésolution, utilise la même méthode d'optimisation sur la grille fine et sur la grille grossière, avec les mêmes paramètres. Cela permet de tester l'algorithme en lui-même et non pas la méthode de minimisation associée.

Les deux principaux algorithmes multigrilles, adaptés aux non-linéarités, ont été présentés dans la section 3.3 du chapitre sur les méthodes multigrilles. Il s'agit de la méthode FAS et de la méthode N-MG. Au lieu d'utiliser l'algorithme N-MG, nous étudions une variante que l'on nomme GN-MG pour Gauss Newton- Multigrille. En effet, le GN-MG consiste à appliquer les méthodes

multigrilles sur une méthode de Gauss-Newton, couramment appelée méthode incrémentale en assimilation de données.
Nous décrivons ensuite la méthode multi-incrémentale.
Nous nous limitons, ici encore, à deux grilles et utilisons les mêmes notations, Ω^f pour la grille fine et Ω^c pour la grille grossière, ainsi que I_c^f et I_f^c pour les transferts inter-grilles. Nous rappelons que l'itéré $\mathbf{x}_0^f$ est transposé sur la grille grossière avec un opérateur de restriction $\hat{I}_f^c$ qui peut être différent de celui transférant le résidu $-\nabla\mathcal{J}^f[\mathbf{x}_0^f]$. Cependant dans la suite, nous utiliserons un unique opérateur I_f^c pour les deux variables.

5.1.1 La méthode FAS

La méthode Full Approximation Scheme n'est pas basée sur une approximation linéaire. Il s'agit d'appliquer la méthode multigrille directement sur le système non linéaire $\nabla\mathcal{J}^f(\mathbf{x}_0^f) = 0$. Nous nous inspirons des équations (3.39) et (3.40) pour décrire l'équation résiduelle que le FAS résout sur la grille grossière :

$$\nabla\mathcal{J}^c_{\text{FAS}}(\mathbf{x}_0^c) = \nabla\mathcal{J}^c(\mathbf{x}_0^c) + I_f^c\nabla\mathcal{J}^f(\mathbf{x}_0) - \nabla\mathcal{J}^c(I_f^c\mathbf{x}_0^f) = 0 \tag{5.2}$$

ce qui revient à minimiser la nouvelle fonction coût $\mathcal{J}^c_{\text{FAS}}$:

$$\mathcal{J}^c_{\text{FAS}}(\mathbf{x}_0^c) = \mathcal{J}^c(\mathbf{x}_0^c) + < L, \mathbf{x}_0^c > \tag{5.3}$$

si l'on définit

$$L = I_f^c\nabla\mathcal{J}^f(\mathbf{x}_0) - \nabla\mathcal{J}^c(I_f^c\mathbf{x}_0^f) \tag{5.4}$$

Nous détaillons l'algorithme qui est aussi schématisé figure 5.1.

Algorithme de la méthode FAS :
Minimisation de $\mathcal{J}^f(\mathbf{x}_0^f)$, fonction coût (5.1) discrétisée sur la maille Ω^f :

Données : $\mathbf{x}_{0,k=0} = \mathbf{x}^b$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. On fait ν_1 itérations de minimisation de la fonction coût $\mathcal{J}^f[\mathbf{x}_0^f]$ sur la grille fine ; à partir de $\mathbf{x}_{0,k}^f$, pour trouver le nouvel itéré $\mathbf{x}_0^f$.
2. On se place à basse résolution pour minimiser la nouvelle fonction coût $\mathcal{J}^c_{\text{FAS}}[\mathbf{x}_0^c]$ (5.3) ; à partir de $\mathbf{x}_0^c = I_f^c\mathbf{x}_0^f$.

$$\mathcal{J}^c_{\text{FAS}}[\mathbf{x}_0^c] = \mathcal{J}^c[\mathbf{x}_0^c] - < \nabla\mathcal{J}^c[I_f^c\mathbf{x}_0^f] - I_f^c\nabla\mathcal{J}^f[\mathbf{x}_0^f], \mathbf{x}_0^c > \tag{5.5}$$

3. On calcule le nouvel itéré de la grille fine : $\mathbf{x}^f_{0,\,\text{afterCGC}} = \mathbf{x}_0^f + I_c^f(\mathbf{x}_0^c - I_f^c\mathbf{x}_0^f)$.
4. On fait ν_2 itérations de minimisation de la fonction coût $\mathcal{J}^f[\mathbf{x}_0^f]$ sur la grille fine, pour trouver l'itéré $\mathbf{x}_{0,k+1}^f$
5. $k = k + 1$

Fin de la Boucle externe.

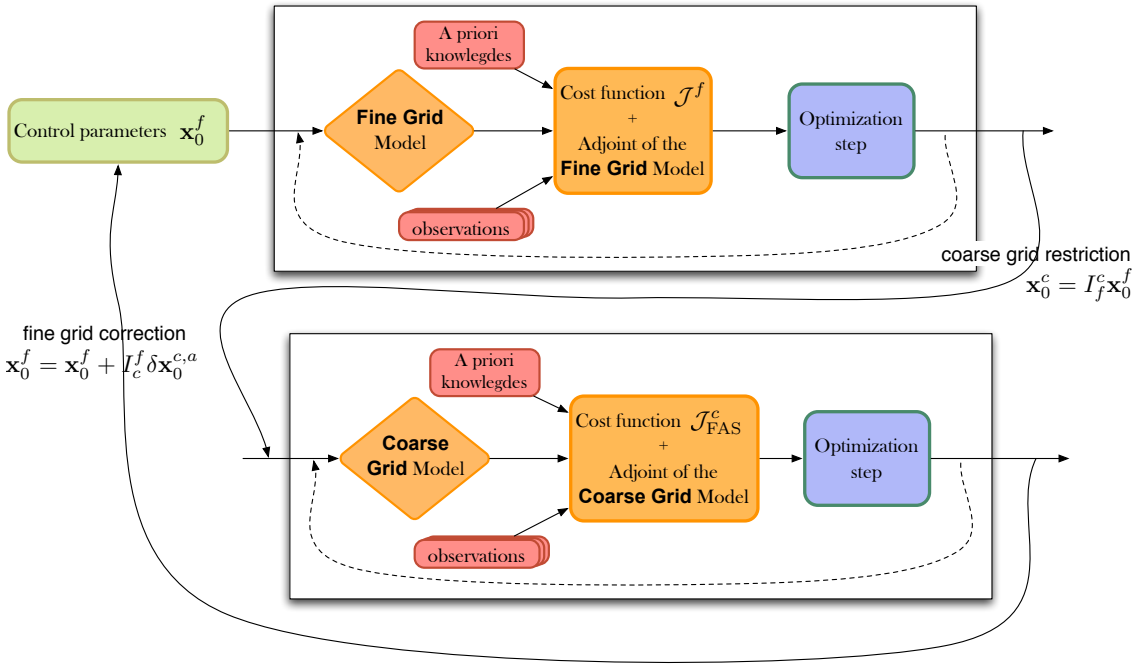

FIG. 5.1 – L'algorithme FAS pour l'assimilation de données

5.1.2 La méthode Gauss Newton - multigrille (GN-MG)

Dans le chapitre sur la résolution du système optimal de l'assimilation de données, est présenté l'algorithme de la méthode incrémentale (voir paragraphe 2.2). La méthode incrémentale remplace la minimisation d'une fonctionnelle non linéaire par la minimisation d'une séquence de fonctions coût quadratiques.
Nous nous inspirons des méthodes Newton-Multigrille, présentées dans le chapitre sur les méthodes multigrilles (voir paragraphe 3.3.2), pour appliquer les méthodes multigrilles à la minimisation de chacune des fonctions coût quadratiques. Cet algorithme portera le nom de Gauss Newton-Multigrille, ou GN-MG.
Chaque fonction coût à minimiser peut s'écrire (rappel équation(2.3)) :

$$\mathcal{J}(\delta\mathbf{x}_0) = \frac{1}{2}\sum_{i=0}^{N-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i]^T\mathbf{R}_i^{-1}[\mathbf{H}_{t_i}\mathbf{M}_{0,i}\delta\mathbf{x}_0 - \mathbf{d}_i] + \frac{1}{2}(\delta\mathbf{x}_0 - \delta\mathbf{x}^b)^T\mathbf{B}^{-1}(\delta\mathbf{x}_0 - \delta\mathbf{x}^b) \quad (5.6)$$

avec

$$\mathbf{d}_i = \mathbf{y}_{t_i} - H_{t_i}(M_{0,i}(\mathbf{x}_{0,k})), \quad \delta\mathbf{x}^b = \mathbf{x}^b - \mathbf{x}_{0,k} \quad (5.7)$$

Le vecteur d'innovation $\mathbf{d}_i$ et la pseudo-ébauche $\delta\mathbf{x}^b$ sont remis à jour, après minimisation de $\mathcal{J}(\delta\mathbf{x}_0)$ au sein de ce que l'on nomme une boucle externe.

Pour la minimisation de chaque fonction quadratique (5.6), l'algorithme des multigrilles est appliqué et permet de résoudre $\nabla\mathcal{J}^f[\delta\mathbf{x}_0^f] = 0$ sur la grille fine Ω^f. Nous devons donc résoudre sur la grille grossière Ω^c l'équation résiduelle suivante :

$$\nabla\mathcal{J}^c[\delta\mathbf{x}_0^c] = \nabla\mathcal{J}^c[I_f^c\delta\mathbf{x}_0^f] - I_f^c\nabla\mathcal{J}^f[\delta\mathbf{x}_0^f] \quad (5.8)$$

puis corriger

$$\delta \mathbf{x}^f_{0,\text{afterCGC}} = \delta \mathbf{x}^f_0 + I^f_c(\delta \mathbf{x}^c_0 - I^c_f \delta \mathbf{x}^f_0) \tag{5.9}$$

Parce que la fonction coût $\mathcal{J}^c[\delta \mathbf{x}^c_0]$ est quadratique, l'équation (5.8) peut également s'écrire :

$$\mathcal{H}^c(\delta \mathbf{x}^c_0 - I^c_f \delta \mathbf{x}^f_0) = -I^c_f \nabla \mathcal{J}^f(\delta \mathbf{x}^f_0) \tag{5.10}$$

Le modèle de la grille grossière est, comme celui de la grille fine, linéarisé autour d'une trajectoire calculée par le modèle non linéaire. Deux possibilités s'offrent à nous ; linéariser autour de $\mathbf{x}^c_{0,k}$ après intégration d'un modèle non linéaire à basse résolution, ou linéariser autour de $I^c_f \mathbf{x}^f_{0,k}$. A priori, les deux méthodes sont valables, mais nous choisissons la deuxième pour faire en sorte que le modèle linéaire grossier soit au plus proche de la solution de la grille fine.
Donc, sauf mention contraire, le modèle de la grille grossière est linéarisé autour de la solution $I^c_f \mathbf{x}^f_{0,k}$.

Nous présentons l'algorithme complet de la méthode, dans le cadre de V-cycles FAS, ainsi qu'un schéma explicatif figure 5.2.

Algorithme de la méthode GN-MG :
Minimisation de $\mathcal{J}^f(\mathbf{x}^f_0)$, fonction coût (5.1) discrétisée sur la maille Ω^f :

Données : $\mathbf{x}_{0,k=0} = \mathbf{x}^b$ est l'estimation actuelle.
Boucle sur k jusqu'à convergence,

1. $\mathbf{d}_i = \mathbf{y}_{t_i} - H^f_{t_i}(M^f_{0,i}(\mathbf{x}^f_{0,k})), \quad \delta \mathbf{x}^{bf} = \mathbf{x}^{bf} - \mathbf{x}^f_{0,k}$
2. Minimisation de $\mathcal{J}^f(\delta \mathbf{x}^f_0)$ en partant de $\delta \mathbf{x}^f_{0,j=0} = 0$:

$$\mathcal{J}^f[\delta \mathbf{x}^f_0] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}^f_{t_i}\mathbf{M}^f_{0,i}\delta \mathbf{x}^f_0 - \mathbf{d}_i\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\|\delta \mathbf{x}^f_0 - \delta \mathbf{x}^{bf}\|^2_{(\mathbf{B}^f)^{-1}} \tag{5.11}$$

   (a) *Boucle interne :* On fait ν_1 itérations de minimisation de la fonction coût $\mathcal{J}^f[\delta \mathbf{x}^f_0]$ sur la grille fine, pour trouver le nouvel incrément $\delta \mathbf{x}^f_0$, à partir de $\delta \mathbf{x}^f_{0,j}$
   (b) On se place à basse résolution pour minimiser la nouvelle fonction coût $\mathcal{J}^c_{\text{FAS}}[\delta \mathbf{x}^c_0]$ à partir de $\delta \mathbf{x}^c_0 = I^c_f \delta \mathbf{x}^f_0$

$$\mathcal{J}^c_{\text{FAS}}[\delta \mathbf{x}^c_0] = \mathcal{J}^c[\delta \mathbf{x}^c_0] - < \nabla \mathcal{J}^c[I^c_f \delta \mathbf{x}^f_0] - I^c_f \nabla \mathcal{J}^f[\delta \mathbf{x}^f_0], \delta \mathbf{x}^c_0 > \tag{5.12}$$

   (c) On calcule le nouvel incrément de la grille fine : $\delta \mathbf{x}^f_{0,\text{afterCGC}} = \delta \mathbf{x}^f_0 + I^f_c(\delta \mathbf{x}^c_0 - I^c_f \delta \mathbf{x}^f_0)$.
   (d) On fait ν_2 itérations de minimisation de la fonction coût $\mathcal{J}^f[\delta \mathbf{x}^f_0]$ sur la grille fine, à partir de $\delta \mathbf{x}^f_{0,\text{afterCGC}}$, pour trouver le nouvel incrément $\delta \mathbf{x}^f_{0,j+1}$
   (e) On revient éventuellement au début de la boucle interne. $j \leftarrow j+1$.
3. Corriger la solution,

$$\mathbf{x}^f_{0,k+1} = \mathbf{x}^f_{0,k} + \delta \mathbf{x}^f_{0,j} \tag{5.13}$$

4. $k = k+1$

Fin de la Boucle externe.

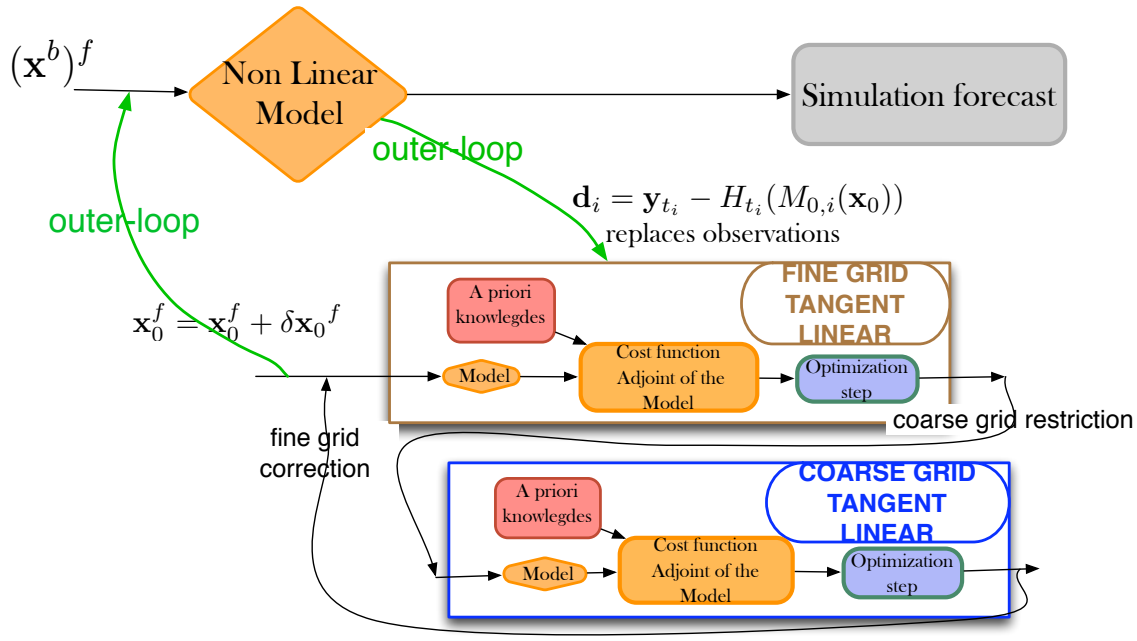

FIG. 5.2 – L'algorithme GN-MG pour l'assimilation de données

5.1.3 La méthode multi-incrémentale

La méthode incrémentale est couramment utilisée en assimilation de données pour l'océanographie ou la météorologie. Cependant, le coût de calcul reste trop élevé pour qu'elle soit utilisée telle quelle, en mode opérationnel. C'est pourquoi, dans ce genre d'expérimentations, c'est une variante appelée méthode incrémentale réduite qui est implémentée. Cette méthode est basée sur la résolution du système optimal linéarisé dans un espace simplifié à l'aide d'un opérateur de simplification (voir chapitre 2, section 2.2).

La méthode de simplification qui nous intéresse plus particulièrement a été introduite par Veersé et Thépaut [63] pour des applications opérationnelles spécifiques à Météo France. Il s'agit d'utiliser un espace de dimension réduite comme espace simplifié. Cette méthode est nommée multi-incrémentale car elle consiste à utiliser plusieurs niveaux de simplifications, chacune des simplifications étant associée à une réduction de la taille de discrétisation. Dans ce cas, l'opérateur de simplification (S, dans la section 2.2) correspond à un opérateur de restriction ; qui est éventuellement associé à une simplification du modèle. Comme il s'agit d'une méthode qui résout le problème optimal de l'assimilation de données dans des espaces de différentes dimensions, et qui est, de plus, utilisée en mode opérationnel, nous allons la comparer avec la méthode multigrille. Nous nous plaçons dans le cas où il n'est utilisé que deux résolutions. La figure 5.3 représente cette configuration. Voici l'algorithme dans le cas où $S = I_f^c$, en considérant que la fonction coût est résolue sur l'espace Ω^f et que l'espace réduit est Ω^c :

Algorithme de la méthode multi-incrémentale :
Minimisation de $\mathcal{J}^f(\mathbf{x}_0^f)$, fonction coût (5.1) discrétisée sur la maille Ω^f :

Données : $\mathbf{x}_{0,k=0} = \mathbf{x}^b$ est l'estimation actuelle.

Boucle sur k jusqu'à convergence,

1. $\mathbf{d}_i = \mathbf{y}_{t_i} - H^f_{t_i}(M^f_{0,i}(\mathbf{x}^f_{0,k})), \quad \delta\mathbf{x}^{bf} = \mathbf{x}^{bf} - \mathbf{x}^f_{0,k}$
2. Minimisation de $\mathcal{J}^f(\delta\mathbf{x}^f_0)$ en partant de $\delta\mathbf{x}^f_{0,j=0} = 0$:

$$\mathcal{J}^f[\delta\mathbf{x}^f_0] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}^f_{t_i}\mathbf{M}^f_{0,i}\delta\mathbf{x}^f_0 - \mathbf{d}_i\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\|\delta\mathbf{x}^f_0 - \delta\mathbf{x}^{bf}\|^2_{(\mathbf{B}^f)^{-1}} \tag{5.14}$$

   (a) *Boucle interne :* On se place à basse résolution et on minimise $\mathcal{J}^c[\delta\mathbf{x}^c_0]$:

$$\mathcal{J}^c[\delta\mathbf{x}^c_0] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}^c_{t_i}\mathbf{M}^c_{0,i}\delta\mathbf{x}^c_0 - \mathbf{d}_i\|^2_{\mathbf{R}_i^{-1}} + \frac{1}{2}\|\delta\mathbf{x}^c_0 - I^c_f\delta\mathbf{x}^f_b\|^2_{(\mathbf{B}^c)^{-1}} \tag{5.15}$$

   (b) On calcule le nouvel incrément de la grille fine : $\delta\mathbf{x}^f_{0,j} = I^f_c\delta\mathbf{x}^c_0$
3. Corriger la solution,

$$\mathbf{x}^f_{0,k+1} = \mathbf{x}^f_{0,k} + \delta\mathbf{x}^f_{0,j} \tag{5.16}$$

4. $k = k + 1$

Fin de la Boucle externe.

L'opérateur $\mathbf{H}^c_{t_i}$ est équivalent au $\mathbf{H}_{t_i}S^{-I}$ présenté dans la section 2.2.

La méthode multi-incrémentale correspond donc à une méthode GN-MG pour laquelle on aurait omis les itérations à haute résolution, c'est-à-dire avec $\delta\mathbf{x}^f_{0,j=0} = 0$ lors de l'itération 2.b. et $\delta\mathbf{x}^f_{0,j+1} = \delta\mathbf{x}^f_{0,\text{afterCGC}}$ lors de l'itération 2.d. de l'algorithme de la méthode GN-MG. Le terme $< \nabla\mathcal{J}^c[I^c_f\delta\mathbf{x}^f_0] - I^c_f\nabla\mathcal{J}^f[\delta\mathbf{x}^f_0], \delta\mathbf{x}^c_0 >$ est alors nul.
Sauf mention contraire, on utilise le même modèle sur la grille grossière que celui de la méthode GN-MG. C'est-à-dire qu'on linéarise autour de la solution haute résolution $I^c_f\mathbf{x}^f_{0,k}$.

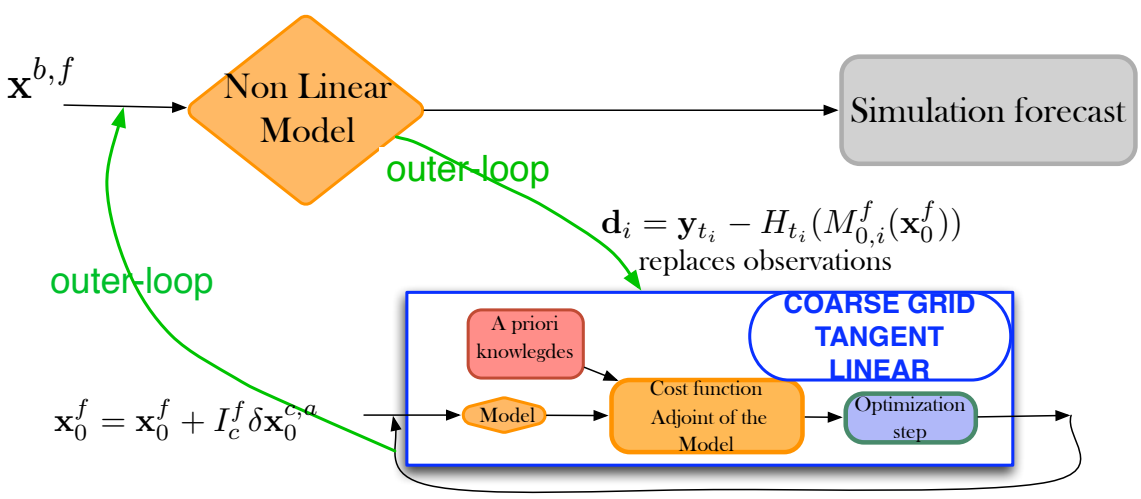

FIG. 5.3 – L'algorithme multi-incrémental pour l'assimilation de données

5.1.4 Les contraintes d'utilisation

Nous avons présenté, pour le cas linéaire, diverses contraintes à vérifier pour que les méthodes multigrilles soient efficaces. Dans le cas non linéaire, aux conditions de convergence des méthodes multigrilles, peuvent s'ajouter selon les cas, les conditions de convergence des méthodes incrémentales.

Contraintes communes aux méthodes multigrilles, FAS et GN-MG

- L'ellipticité du Hessien $\mathcal{H}$ est à vérifier pour être en mesure de trouver une méthode itérative qui ait la propriété de lissage.
  L'ellipticité a été étudiée en détail dans le chapitre précédent, sur le cas linéaire.
  Dans le cas non linéaire, le Hessien n'est plus un opérateur linéaire, étudier son ellipticité est alors moins pertinent. Cependant l'ellipticité n'est pas une condition nécessaire de convergence.

- La méthode d'optimisation doit vérifier la propriété de lissage. Nous utilisons une méthode de minimisation robuste, bien adaptée aux non linéarités, le M1QN3 [21]. La méthode M1QN3 a les mêmes propriétés fréquentielles que la méthode de gradient conjugué dont elle est dérivée. Nous avons explicité dans le chapitre précédent (4.2.2), en reprenant les remarques de Nash [40], que les méthodes de gradient conjugué ne sont pas garanties d'être des méthodes lissantes. Pourtant elles sont dans leur construction, plus à même de réduire les hautes fréquences de l'erreur. Il en est de même pour la méthode de minimisation M1QN3.

- Les opérateurs de transfert et les opérateurs de la grille grossière doivent vérifier la propriété d'approximation.
  Pour chaque système linéaire résolu par l'algorithme GN-MG, Les résultats de convergence sont les mêmes que pour ceux de la section 4.2.3. Nous avons accès, théoriquement, au facteur de convergence d'un passage par la grille grossière (voir équation (4.7)). Nous savons aussi que, pour chaque Hessien linéarisé globalement, le Hessien grossier doit être proche du Hessien fin en approchant la relation de Galerkin :

$$\|\mathcal{H}^c - I_f^c \mathcal{H}^f I_c^f\| \leq (\Delta x^f)^\alpha \tag{5.17}$$

  avec α donnée par la propriété de lissage (3.26). Le coefficient α' est en général supérieur ou égal à l'ordre de consistance des discrétisations.
  En ce qui concerne l'algorithme FAS, l'étude du facteur de convergence d'un passage par la grille grossière est quelque peu différente (voir section 3.3.3). Mais les conclusions sont les mêmes. Le facteur de convergence est fortement dépendant de la qualité d'approximation effectuée à basse résolution du système haute résolution. Une bonne approximation peut être construite en approchant la relation de Galerkin (5.17).
  Dans ce cas, étant donné l'expression du Hessien :

$$\mathcal{H} = \sum_i (\mathbf{H}_{t_i}\mathbf{M}_{0,i})^T \, \mathbf{R}^{-1}\mathbf{H}_{t_i}\mathbf{M}_{0,i} + \mathbf{B}^{-1} + \sum_i \left(\frac{\partial^2 H_{t_i} M_{0,i}}{\partial \mathbf{x}_0^2}\right)^T \mathbf{R}^{-1}[H_{t_i}(M_{0,i}(\mathbf{x}_0)) - \mathbf{y}_{t_i}], \tag{5.18}$$

le Hessien $\mathcal{H}^c$ approche (5.17) si il existe une constante c tel que $(I_c^f)^T = c\, I_f^c$, et si les opérateurs vérifient les approximations suivantes :

$$\begin{cases} \|H_{t_i}^c M_{0,i}^c - H_{t_i}^f M_{0,i}^f I_c^f\| \leq (\Delta x^f)^\alpha, \quad \forall i \\ \|(\mathbf{R}^c)^{-1} - \frac{1}{c}(\mathbf{R}^f)^{-1}\| \leq (\Delta x^f)^\alpha \\ \|(\mathbf{B}^c)^{-1} - I_f^c(\mathbf{B}^f)^{-1} I_c^f\| \leq (\Delta x^f)^\alpha \end{cases} \tag{5.19}$$

Nous constatons également que dans le Hessien, apparaissent les observations et plus particulièrement les différences entre le modèle et les observations. Elles ont donc un rôle dans la minimisation, sûrement plus important que dans le cas d'un modèle linéaire, où le Hessien est indépendant des observations.
Bien utilisées, les méthodes FAS convergent vers la solution $\mathbf{x}_0^{f,a}$ du système optimal haute résolution.

Pour le GN-MG

Si l'on résout jusqu'à convergence au sein de chaque boucle interne, alors on obtient les résultats de convergence de la méthode incrémentale (Gauss-Newton), déjà explicités section 2.2. Selon Gratton *et al.* ([23]), la méthode Gauss-Newton ou méthode incrémentale converge vers la solution $\mathbf{x}_0^{f,a}$ localement, c'est-à-dire si $\mathbf{x}^{f,b}$ est suffisamment proche de $\mathbf{x}_0^{f,a}$.
Pour qu'il y ait convergence de la méthode incrémentale, il faut donc :

$$\|\mathbf{x}^{f,b} - \mathbf{x}_0^{f,a}\| \leq \epsilon, \tag{5.20}$$

Le problème étant qu'on ne connaît pas par avance la solution optimale $\mathbf{x}_0^{f,a}$, il est difficile de vérifier cette condition. Supposer qu'il y a peu de non linéarités, nous permet de supposer que la fonction est presque quadratique et qu'il y a donc peu de chances que la méthode converge vers un minimum local.

Pour la méthode multi-incrémentale

La convergence de la méthode incrémentale réduite, dont la méthode multi-incrémentale est une variante, est explicitée section 2.2. La méthode est basée sur la même méthode Gauss-Newton que le GN-MG. La convergence dépend fortement de la qualité de l'ébauche qui doit être proche de la solution.
Cependant, il est évident que le problème résolu (uniquement à basse résolution dans notre cas) n'est pas le même et donc que la solution de convergence n'est pas $\mathbf{x}_0^{f,a}$. Dans le cas d'une méthode multi-incrémentale, la qualité d'approximation de $\mathcal{H}^c$ est importante pour la convergence de la méthode mais influe également sur la solution optimale calculée.

5.2 Spécificité des tests non linéaires

5.2.1 Le modèle et les observations

Nous travaillons avec une équation de Burgers non visqueuse, dans un domaine à une dimension $\Omega = [0, L]$, du temps 0 à T.

$$\partial_t \mathbf{x} + \frac{1}{2}\partial_x \mathbf{x}^2 = 0 \tag{5.21}$$

avec des conditions aux bords périodiques.
La méthode de discrétisation choisie est, comme pour l'exemple linéaire, une méthode en différences finies sur un maillage régulier, découpé en N_x intervalles d'espace de taille Δx et N_t intervalles de temps de durée Δt. Le schéma de discrétisation utilisé est le schéma Lax- Friedrichs. Ce schéma a l'avantage d'être stable et différentiable mais néamoins très diffusif. Il est basé sur le calcul d'une valeur intermédiaire, au temps "fictif" $n+\frac{1}{2}$, notée $\mathbf{x}^j_{n+\frac{1}{2}}$ pour ensuite calculer la valeur au temps n, $\mathbf{x}^j_n$. On rappelle que $\mathbf{x}^j_n$ est la valeur approchée de $\mathbf{x}$ au point $\mathbf{x}(j\Delta x, n\Delta t)$. Les équations discrètes sont :

$$\left\{\begin{array}{rcl} \mathbf{x}^j_{n+\frac{1}{2}} & = & \frac{1}{2}\left(\mathbf{x}^j_n + \mathbf{x}^{j+1}_n\right) - \frac{\Delta t}{2\Delta x}\left(\frac{1}{2}(\mathbf{x}^{j+1}_n)^2 - \frac{1}{2}(\mathbf{x}^j_n)^2\right) \\ \mathbf{x}^j_{n+1} & = & \frac{1}{2}\left(\mathbf{x}^{j-1}_{n+\frac{1}{2}} + \mathbf{x}^j_{n+\frac{1}{2}}\right) - \frac{\Delta t}{2\Delta x}\left(\frac{1}{2}(\mathbf{x}^j_{n+\frac{1}{2}})^2 - \frac{1}{2}(\mathbf{x}^{j-1}_{n+\frac{1}{2}})^2\right) \end{array}\right. , \forall j \in [0, N_x] \tag{5.22}$$

Comme dans le cas linéaire, on utilise l'expression générale de la solution $\mathbf{x}_i$ au temps i :

$$\mathbf{x}_i = M_{i-1,i}(\mathbf{x}_{i-1}) = M_{0,i}(\mathbf{x}_0) \tag{5.23}$$

avec $M_{0,i} = M_{i-1,i} \circ M_{i-2,i-1} \circ \ldots \circ M_{0,1}$.

Il s'agit d'expériences jumelles ; les observations artificielles sont générées par le modèle de la grille fine à partir d'un état initial connu. Ceci nous permet de connaître la solution vraie, et de vérifier que la solution optimale de l'assimilation de données s'en approche :

$$\mathbf{y}_i = M_{0,i}(\mathbf{x}^t_0) + \mathcal{G}(0, \sigma_0^2) \tag{5.24}$$

où $\mathcal{G}(0, \sigma_0^2)$ est un vecteur gaussien, qui simule le bruit des observations réelles.

5.2.2 La fonction coût

Les matrices de covariances d'erreurs des observations et de l'ébauche $\mathbf{R}_i^{-1}$ et $\mathbf{B}^{-1}$ sont définies de la même façon que dans le cas test linéaire du chapitre précédent (voir section 4.3.1). Nous choisissons une matrice $\mathbf{B}^{-1}$ qui approche une gaussienne. Nous observons directement $\mathbf{x}$. L'opérateur d'observation $H(.)$ est, ici aussi, l'identité. La fonction coût discrète de l'assimilation de données devient :

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2\sigma_o^2}\sum_{i=0}^{N-1}\|\mathbf{M}_{0,i}\mathbf{x}_0 - \mathbf{y}_i\|^2 + \frac{\beta}{2\sigma_b^2}\left\|\left(\mathbf{I} - \frac{\sigma_b^2}{4}\triangle\right)(\mathbf{x}_0 - \mathbf{x}^b)\right\|^2 \tag{5.25}$$

Pour toutes les méthodes de résolution étudiées, nous utilisons une méthode d'optimisation de type gradient conjugué, adaptée aux non linéarités, le M1qn3, avec à chaque utilisation les mêmes réglages.

5.2.3 Les deux grilles

Tout comme dans le cas linéaire, on utilise deux discrétisations avec un raffinement d'ordre 2 en espace et en temps.
Les deux modèles, que ce soit sur la grille fine ou la grille grossière, sont discrétisés selon le même schéma aux différences finies. Seuls les paramètres, Δt et Δx, de la taille de la maille, changent.

Pour ce qui est de la fonction coût, on utilise, là aussi, les mêmes paramètres pour les deux grilles (y compris les mêmes matrices d'erreurs) ; avec néanmoins un petit changement. En effet si on considère qu'on a accès aux observations tous les n_t^f pas de temps sur la grille fine, alors on aura accès aux observations tous les $n_t^c = n_t^f/2$ pas de temps sur la grille grossière, car le raffinement est d'ordre 2. De même pour n_x^f et n_x^c. La dimension de l'espace de contrôle varie donc d'une grille à l'autre mais le nombre d'observations reste le même sur les deux grilles.

Pour le transfert en espace, entre les grilles, on utilise, que ce soit pour le résidu ou l'état, les mêmes opérateurs de full-weighting et interpolation linéaire, qui sont ceux également utilisés dans le cas linéaire (4.39). L'ébauche de la grille grossière est transférée de la grille fine par l'opérateur de restriction

$$\mathbf{x}^{c,\,b} = I_f^c \mathbf{x}^{f,\,b}$$

Le transfert en temps n'intervient que pour le transfert de la solution $\mathbf{x}_i^f$ sur la grille grossière. Dans le cas du GN-MG et du multi-incrémental (voir sections 5.1.2 et 5.1.3), ce transfert est utilisé pour le transport de la variable du modèle linéaire tangent. Nous optons pour un transfert simple : ne prendre qu'un point sur deux en temps, sans faire de moyenne.

5.2.4 Paramètres et configuration

Le modèle est intégré sur le domaine $[0, L = 1]$, du temps 0 à $T = 0.512$.
Sur la grille fine Ω^f, on calcule les états pour 204800 points distants de $\Delta x^f = 0.0025$ en espace et $\Delta t^f = 0.001$ en temps.
Sur la grille grossière, les facteurs de raffinement en temps et en espace sont égaux à 2.
Pour créer les observations (5.24), on intègre le modèle sur la grille fine à partir de l'état vrai suivant :

$$\mathbf{x}_0^t(x) = \sin(2\pi x/L) \tag{5.26}$$

bruité par une gaussienne de variance $\sigma_o^2 = 0.02$.
Les paramètres de la fonction coût sont les suivants : $\sigma_b = 0.2$, $\beta = 1$. Nous rappelons l'étude sur l'ellipticité section 4.3.2 pour laquelle une valeur de $\beta = 100$ nous donnait $\gamma = \frac{\beta\sigma_b^2}{T\sigma_o^2} = 1$ et un Hessien elliptique. Par rapport au cas linéaire, seul le modèle est modifié. Cependant changer le modèle modifie également la valeur de γ_{lim}, pour laquelle le Hessien est elliptique. Sans connaître précisément γ_{lim}, nous avons choisi γ de telle sorte que l'état initial optimal soit suffisamment proche de l'état vrai.
400 points d' observations sont pris en compte. Ce qui correspond à considérer une observation, tous les $n_t^f = 32$ pas de temps et $n_x^f = 16$ pas d'espace sur la grille fine ; et donc sur la grille grossière tous les $n_t^c = 16$ pas de temps et $n_x^c = 8$ pas d'espace.
L'ébauche, ou l'état initial avant optimisation, est prise égale à :

$$\mathbf{x}^b(x) = 0.9\sin(2\pi x/L + \pi) + 0.05\sin(5.2\pi x/L) \tag{5.27}$$

Elle est constituée de l'état vrai avec un facteur d'amplification et un déphasage de π. Ce sont deux manières de bruiter l'état vrai pour s'en éloigner suffisamment.
Nous intégrons également dans l'ébauche un terme haute fréquence, alors que la solution vraie est en basse fréquence. Ceci permet d'introduire des hautes fréquences supplémentaires (en plus du bruit ajouté aux observations) pour se placer dans un cas où intégrer le modèle non linéaire à haute résolution apporte de l'information supplémentaire par rapport à la basse résolution.

5.3 Les multigrilles éprouvées

Nous présentons ici les résultats des différents algorithmes. Nous commençons par les décrire ainsi que les diagnostics utilisés. Ensuite, nous vérifions la validité des hypothèses de linéarité et des approximations faites par la grille grossière.
Enfin, nous étudions la configuration des méthodes multigrilles. Après avoir fixé les différents paramètres inhérents aux algorithmes, nous pouvons comparer les résultats des méthodes multigrilles avec les méthodes monogrilles et le multi-incrémental.

5.3.1 Les variables d'initialisation et les diagnostics utilisés

La méthode de référence est la solution à convergence du problème 4D-VAR non linéaire sur la grille Ω^f. C'est ainsi que l'on obtient l'état initial optimal et son évolution dans le temps, présenté sur la Figure 5.4. Nous observons qu'un choc se produit à peu près entre t=0.2 et t=0.3 puis qu'il se dissipe. Nous comparons la méthode FAS à cette méthode de référence.

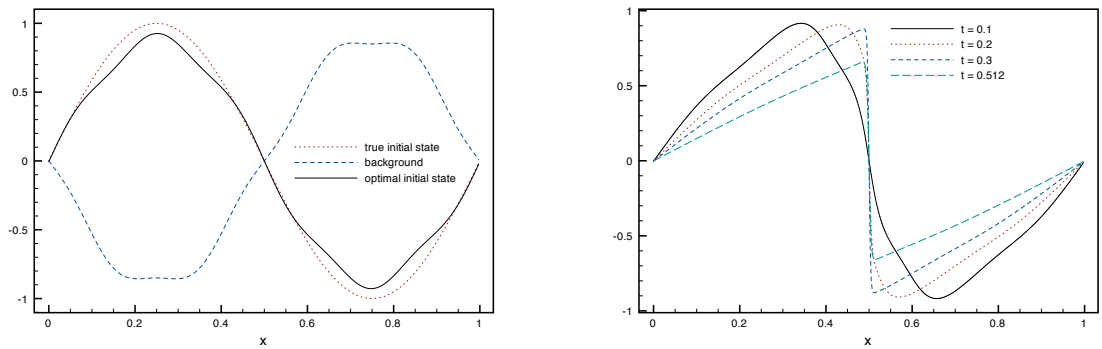

FIG. 5.4 – État initial vrai, ébauche et état initial optimal, à gauche. À droite, l'évolution dans le temps de l'état optimal

Notre deuxième méthode de référence est la méthode incrémentale, qui utilise l'hypothèse linéaire tangent au sein de chaque boucle externe. Pour chaque minimisation, le problème quadratique est résolu à convergence. Cette méthode est comparée à la méthode multigrille GN-MG et à la méthode multi-incrémentale.

Nous allons considérer, au cours de la minimisation, l'évolution de l'erreur RMS de l'état optimal par rapport à l'état vrai, $RMS(\mathbf{x}, \mathbf{x}^t)$ avec :

$$RMS(\mathbf{x}, \mathbf{x}^t) = \sqrt{\frac{1}{N_x.N_t} \sum_{n=0}^{N_t-1} \sum_{j=0}^{N_x-1} (\mathbf{x}_n^j - (\mathbf{x}_n^j)^t)^2} \tag{5.28}$$

Cette erreur $RMS(\mathbf{x}, \mathbf{x}^t)$ mesure ce que l'on veut réellement minimiser : l'écart entre le modèle $\mathbf{x}$ et les observations non bruitées.

Cette valeur RMS va décroître au fur et à mesure de la minimisation de la fonction coût et peut être calculée quelle que soit la méthode et la fonction coût, c'est-à-dire même si la solution

x a été obtenue après intégration du modèle linéaire. Elle nous permet donc de comparer les différents algorithmes entre eux. Cependant pour la visualiser, il convient de choisir pour abscisse le temps de calcul $T_{\text{computational}}$ du programme. D'autres abscisses telles le nombre d'itérations de la grille fine ou le nombre de cycles effectués donnent des courbes dépendantes de l'algorithme. Il n'est alors pas possible de prendre en compte un passage sur la grille grossière ou de comparer correctement le FAS avec le GN-MG. Visualiser l'erreur RMS en fonction du temps permet de s'affranchir de ces considérations, et permet également de comparer le temps mis à faire quelques itérations sur la grille fine par rapport à celui mis à résoudre le problème sur la grille grossière.

Avant de passer à l'étude des erreurs d'approximation, nous comparons la décroissance de l'erreur RMS des méthodes de référénce, sur la Figure 5.5. Pour la méthode incrémentale, nous allons à convergence pour chaque minimisation de fonctions coût quadratiques. Ne pas aller à convergence au sein de chaque boucle externe, en fixant un nombre limite d'itérations, ralentit la convergence globale de la minimisation de la fonction coût non quadratique. En réglant bien les paramètres de M1QN3, la méthode incrémentale met autant de temps que la minimisation non quadratique, à atteindre la valeur de convergence. C'est à dire, sur la machine utilisée, pour $T_{\text{computational}} \simeq 200s$.

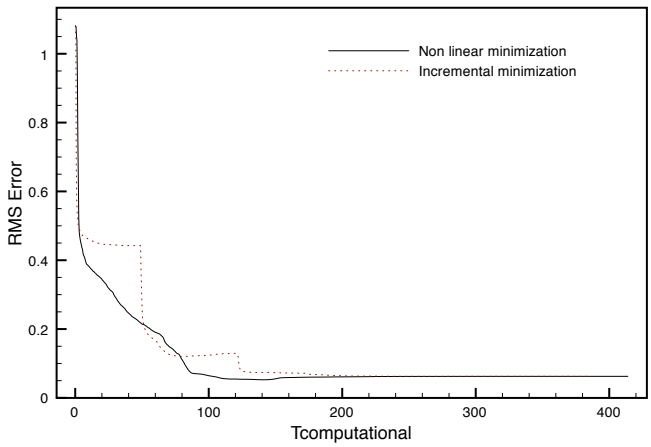

Fig. 5.5 – Décroissance de l'erreur RMS en fonction du temps de calcul en secondes, pour le monogrille et la méthode incrémentale.

5.3.2 Erreurs d'approximation

Hypothèse linéaire tangent

Nous voulons tester ici les conditions de validité de l'hypothèse de linéarité. Cette condition est supposément valide pour un temps d'intégration court.

Sur la Figure 5.6, nous visualisons l'erreur faite par l'hypothèse de linéarisation. C'est-à-dire

que nous avons calculé l'erreur entre le modèle non linéaire et le modèle linéaire :

$$\frac{\|M^f_{0,t}(\mathbf{x}^f_0 + \delta\mathbf{x}^f_0) - M^f_{0,t}(\mathbf{x}^f_0) - \mathbf{M}^f_{0,t}(\delta\mathbf{x}^f_0)\|}{\|\mathbf{M}^f_{0,t}(\delta\mathbf{x}^f_0)\|}(t) \tag{5.29}$$

$\delta\mathbf{x}^f_0$ est l'incrément optimal calculé au sein d'une boucle externe. Si l'hypothèse de linéarisation est valide, alors cette erreur est très petite ($\ll 1$).

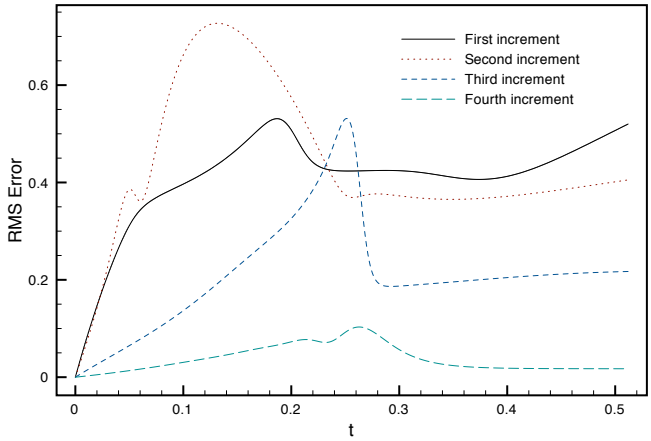

FIG. 5.6 – Les erreurs de linéarisation (5.29) pour la méthode incrémentale, en fonction du temps d'intégration t. Les résultats sont visibles pour les quatre premières boucles externes.

Nous observons que les erreurs tendent vers 0. Ce résultat est logique car au fur et à mesure des boucles externes, le modèle est linéarisé autour d'un état de plus en plus proche de l'état optimal. Le numérateur de (5.29) se rapproche donc de 0.
Nous notons un maximum local pour un temps t proche de 0.2, à partir duquel l'erreur se stabilise. Nous rappelons que ce temps limite représente le choc dû aux non-linéarités. Le maximum est important pour les trois premières boucles externes. Également, les trois premières boucles donnent des erreurs proches de 1, la validité de l'hypothèse linéaire tangent n'est donc pas sure.

Nous avons également calculé les erreurs faites par l'approximation linéaire tangent, cette fois, dans le cas d'incréments calculés par les méthodes GN-MG, sur la Figure 5.7. La méthode GN-MG utilisée comporte des V-cycles, chaque cycle étant composé de 1 itération sur la grille fine, puis une résolution à convergence du problème grossier suivie de 1 itération sur la grille fine. Pour le calcul des erreurs, nous utilisons l'incrément $\delta\mathbf{x}^f_0$ donné par la première minimisation, c'est-à-dire à la sortie de la première boucle interne, après 1 ou 10 cycles de minimisation.

Un cycle de multigrille suffit pour approcher l'incrément optimal. Par contre, après 10 cycles, les erreurs de linéarisation augmentent et s'éloignent des erreurs optimales calculées par l'incrémental monogrille ; au point de violer l'hypothèse de linéarité. Là où la méthode incrémentale converge, le GN-MG diverge. Ceci est dû aux passages par la grille grossière qui, au lieu de diminuer la fonction coût de la grille fine, l'augmentent. Les méthodes GN-MG semblent plus sensibles

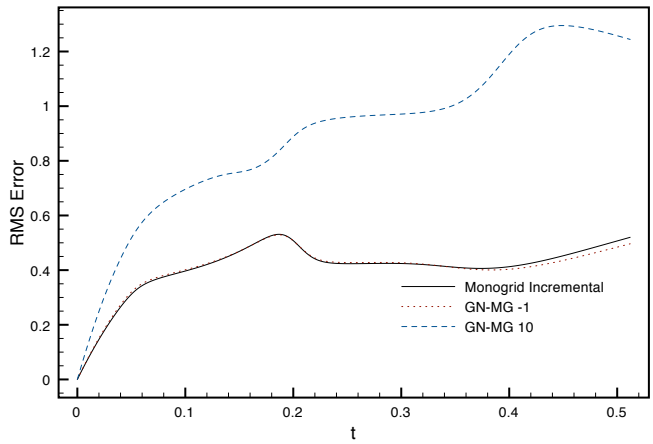

FIG. 5.7 – Les erreurs de linéarisation des méthodes incrémentales et multigrilles en fonction du temps d'intégration t. En trait plein, pour l'incrément optimal calculé par la méthode incrémentale monogrille, en tirets pour l'incrément calculé après 1 cycles de GN-MG (GN-MG -1) et en pointillés après 10 cycles (GN-MG -10).

aux non-linéarités, en particulier la propriété d'approximation ne doit pas être vérifiée dans ce cas. Plusieurs pistes seront données dans la suite, et en particulier dans le chapitre suivant, pour éviter ces divergences. Nous pensons par exemple à l'ajout d'un paramètre de relaxation lors de la correction par grille grossière. Dans ce chapitre, nous nous limiterons au réglage du nombre de cycles. En effet, ne faire qu'un cycle est suffisant pour calculer un incrément valable. Nous vérifierons ce point par la suite.

Approximation grille grossière

Nous nous intéressons ici à l'erreur d'approximation par la grille grossière. Pour cela, nous intégrons un algorithme FAS constitués de V-cycles. Chaque V-cycle comporte 1 première itération sur la grille fine, un passage par la grille grossière où l'on atteint la convergence et une autre itération sur la grille fine.

Après intégration d'un cycle, nous calculons l'erreur entre le modèle grossier et le modèle fin :

$$\frac{\|M^f_{0,t}(\mathbf{x}^f_0) - I^f_c M^c_{0,t}(I^c_f \mathbf{x}^f_0)\|}{\|M^f_{0,t}(\mathbf{x}^f_0)\|}(t) \tag{5.30}$$

pour deux valeurs de $\mathbf{x}^f_0$. La première valeur est prise avant la minimisation sur la grille grossière, et la deuxième après la minimisation, $\mathbf{x}^f_0$ est alors l'état corrigé : $\mathbf{x}^f_{0,\text{afterCGC}} = \mathbf{x}^f_0 + I^f_c(\mathbf{x}^{c,a}_0 - I^c_f \mathbf{x}^f_0)$. Ces erreurs RMS sont visualisées sur la Figure 5.8.

Nous observons que le comportement est relativement similaire aux erreurs dues à l'hypothèse de linéarité (voir la Figure 5.6 et la Figure 5.7) ; sauf aux alentours du maximum local en $t = 0.2$, lors du choc, où les erreurs de linéarisation sont généralement moins stables que les erreurs d'approximation grille grossière.

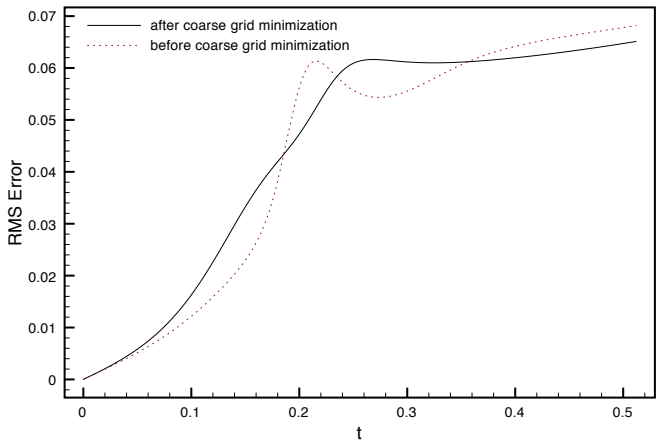

FIG. 5.8 – Les erreurs (5.30) d'approximation par grille grossière des modèles non linéaires, en fonction du temps d'intégration t.

Pour conclure, nous avons également calculé l'erreur entre le modèle grossier et le modèle fin, cette fois, dans le cas linéaire :

$$\frac{\|\mathbf{M}_{0,t}^f \delta\mathbf{x}_0^f - I_c^f \mathbf{M}_{0,t}^c I_f^c \delta\mathbf{x}_0^f\|}{\|\mathbf{M}_{0,t}^f \delta\mathbf{x}_0^f\|}(t) \tag{5.31}$$

Il s'agit donc des modèles intégrés à partir des incréments calculés par les méthodes GN-MG. La méthode GN-MG utilisée comporte des V-cycles similaires à la méthode FAS. Chaque cycle est composé d'une itération sur la grille fine, puis une résolution à convergence du problème grossier suivie d'une itération sur la grille fine. Pour le calcul des erreurs, nous utilisons l'incrément $\delta\mathbf{x}_0^f$ donné par la première minimisation. Ces erreurs sont visualisées sur la Figure 5.9, en tirets avec l'incrément calculé après 5 cycles de multigrilles (GN-MG -5) et en pointillés avec celui calculé après 10 cycles (GN-MG -10). En trait plein, nous retrouvons les erreurs d'approximation du modèle non linéaire de la Figure 5.8, après un passage par la grille grossière.

Un maximum local existe lorsque t approche 0.2s. Nous retrouvons ce maximum sur toutes les courbes. Mais le choc a un effet beaucoup plus marqué sur le GN-MG que sur le FAS.
Les non-linéarités, lorsqu'elles deviennent importantes, entraînent un écart plus grand entre le modèle linéaire fin et le modèle linéaire grossier. Cet écart se retrouve dans les conditions de convergence d'un passage par la grille grossière. Il doit être suffisamment petit pour que la correction par grille grossière ait un impact positif sur la minimisation. Les non-linéarités, en augmentant sensiblement la valeur de cet écart pour le GN-MG, ont un impact négatif sur la convergence des méthodes GN-MG. Les méthodes GN-MG semblent plus sensibles que les méthodes FAS aux non-linéarités.

5.3.3 Paramètres critiques

Dans ce paragraphe, nous étudions les paramètres inhérents aux méthodes multigrilles. Dans les deux cas, FAS et GN-MG, les algorithmes font apparaître divers paramètres dont nous vou-

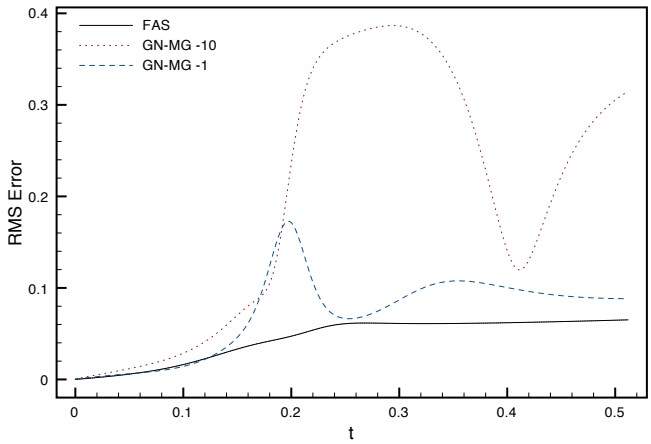

FIG. 5.9 – Les erreurs (5.31) faites par l'approximation du modèle grossier, en fonction du temps d'intégration t. En tirets pour l'incrément calculé après 1 cycle de GN-MG (GN-MG -1) et en pointillés après 10 cycles (GN-MG -10). En trait plein, les erreurs (5.30) du FAS après la correction par la grille grossière.

lons étudier l'influence sur le comportement des méthodes. Nous avons regroupé ces différents paramètres dans le tableau 5.1. Il s'agit du nombre d'itérations sur la grille fine pour chaque cycle, du nombre d'itérations sur la grille grossière pour chaque passage par la grille grossière, et du nombre de cycles de minimisation effectués au sein d'une boucle externe. En gras, sont notés les paramètres intervenant dans les critères de convergence. En effet, les études théoriques sur la convergence des méthodes multigrilles sont basées sur l'hypothèse que l'équation résiduelle, c'est-à-dire le problème grossier, est résolu à convergence. Aussi, la convergence de la méthode incrémentale a été étudiée en résolvant à convergence chaque minimisation.

Paramètre	FAS	GN-MG
Nombre d'itérations de minimisation sur la grille fine (ν_1, ν_2)	×	×
Nombre d'itérations sur la grille grossière	×	×
Nombre de cycles par minimisation		×

TAB. 5.1 – Liste des différents paramètres associés aux méthodes multigrilles. En gras, ceux intervenant dans les critères de convergence.

Sur les deux algorithmes, nous avons testé plusieurs configurations, faisant varier chacun des paramètres un à un.
Dans les deux cas, FAS et GN-MG, le nombre d'itérations de minimisation n'est pas influent ; que ce soit sur la grille fine ou sur la grille grossière. Nous entendons par "non influent", le fait que la perturbation de ce nombre ne modifie pas le comportement général.
En ce qui concerne le nombre d'itérations sur la grille fine. L'étude théorique ne donne pas d'indications quant à la définition de ce paramètre. Nous avons testé 3 valeurs différentes pour $\nu_1 = \nu_2$: 1, 10 et 100. Dans tous les cas, la convergence se révèle plus ou moins rapide mais le comportement global reste le même. Ce paramètre sera donc fixé à 1 par la suite, car l'on veut

minimiser le nombre d'itérations sur la grille fine, comme celles-ci sont les plus coûteuses.
Le nombre d'itérations sur la grille grossière, théoriquement, doit être suffisamment grand pour qu'il y ait convergence. Si ce critère n'est pas vérifié, les études de convergence des méthodes multigrilles ne sont plus valables et leur comportement est donc incertain. En pratique nous avons testé là aussi trois cas, en faisant varier le critère d'arrêt de la méthode de minimisation utilisée. Il ressort de ces expériences qu'il n'est pas nécessaire d'aller jusqu'à convergence, cela ne modifie pas le comportement global et n'induit pas de divergence. Cependant, diminuer le nombre d'itérations de la grille grossière a tendance à la ralentir en atténuant l'effet bénéfique de la correction de la grille grossière. Par la suite, le critère d'arrêt de M1QN3 sur la grille grossière,

$$\epsilon = \frac{\|\nabla \mathcal{J}_{k+1}\|}{\|\nabla \mathcal{J}_k\|} \tag{5.32}$$

qui est le rapport entre la norme du gradient à l'itération courante et la norme du gradient initial, est fixé à 10^{-2}. Nous avons choisi ce critère parce qu'il est un bon compromis entre rapidité de convergence et efficacité de la correction de la grille grossière.

Le dernier paramètre, au contraire, modifie le comportement. Ce paramètre est défini uniquement pour les expériences GN-MG. Pour l'algorithme FAS, on ne minimise qu'une seule fonction, la fonction coût non quadratique d'origine. Le nombre de cycles n'est donc pas un paramètre en tant que tel. En théorie, pour la méthode incrémentale monogrille ou le GN-MG, une fonction coût quadratique doit être minimisée au sein de chaque boucle externe, pour trouver l'incrément optimal. Dans le cas des méthodes GN-MG, il faut donc intégrer autant de cycles que voulus ou nécessaires pour atteindre la convergence.

Nous avons testé ici deux configurations, la première que l'on nomme GN-MG -10, effectue 10 cycles par boucle externe.
La deuxième des méthodes GN-MG est un algorithme où seul un cycle de multigrille est effectué pour minimiser chaque fonction coût quadratique. Cette configuration est nommée GN-MG -1.

Sur la Figure 5.10, nous visualisons la décroissance de l'erreur RMS (5.28), en fonction du temps de calcul pour ces deux configurations GN-MG -10, et GN-MG -1. Seul le nombre de cycles par boucle externe varie. Les autres paramètres sont les mêmes pour les deux configurations ($\nu_1 = \nu_2 = 1$, convergence sur la grille grossière). Nous rappelons également le résultat trouvé par la méthode incrémentale ; pour laquelle seul l'effet de la première boucle externe est visible. Pour chaque courbe GN-MG, un symbole représente soit une intégration du modèle linéaire soit une intégration du modèle non linéaire, sur la grille fine. La décroissance de la fonction coût (non représentée ici) donne des résultats similaires pour chaque méthode.

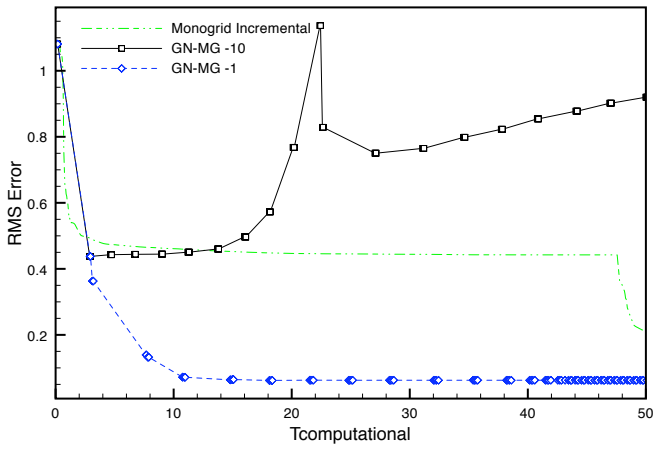

FIG. 5.10 – Décroissance de l'erreur RMS (5.28), selon le temps de calcul, pour les différents algorithmes GN-MG. Les symboles correspondent à un calcul du modèle linéaire ou non linéaire sur la grille fine.

Nous observons que le comportement dépend du nombre de cycle fixé. En effet, pour l'algorithme GN-MG -10, l'erreur RMS commence à diverger lors des derniers cycles de la première boucle externe, c'est à dire pour $T_{\text{computational}} = 15s$. En théorie, il faut aller jusqu'à convergence au sein de chaque boucle externe. Cependant, faire trop de cycles peut dégrader le comportement. Nous remarquons ici que, là où la méthode incrémentale monogrille converge (voir figure 5.5), une méthode multigrille GN-MG peut avoir plus de difficultés. Les algorithmes GN-MG sont donc plus sensibles aux erreurs faites par l'approximation de la grille grossière qu'à celles faites par la linéarisation. Dans ce cas, ne faire qu'un cycle paraît suffisant, et cela semble permettre d'éviter les divergences dues aux non-linéarités.

Dans la suite de ce chapitre, nous réglons donc les différents paramètres selon les résultats obtenus ci-dessus. Pour chaque méthode multigrille, le nombre d'itérations sur la grille fine sera égal à un : $\nu_1 = \nu_2 = 1$; et la convergence sera atteinte pour chaque résolution sur la grille grossière en réglant le critère d'arrêt de M1QN3 (5.32) à 0.01.
Enfin la méthode GN-MG n'effectuera qu'un cycle de multigrille par boucle externe.

5.3.4 Monogrille vs. Multirésolution

On teste ici le fait d'utiliser plusieurs résolutions plutôt qu'une seule. On s'intéresse, à la décroissance de l'erreur $RMS(\mathbf{x}_k, \mathbf{x}^t)$ (5.28) en fonction du temps de calcul. L'erreur RMS est calculée à chaque itération de la méthode de minimisation sur la grille fine ou après correction par grille grossière. Sur la Figure 5.11 est visible la comparaison entre le monogrille (méthode de référence) et le FAS.

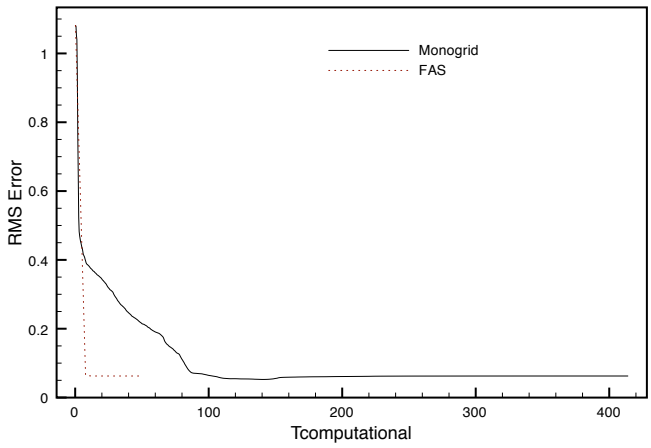

FIG. 5.11 – Décroissance de l'erreur RMS pour le monogrille non linéaire et le FAS, en fonction du temps de calcul en secondes.

Nous comparons ensuite, Figure 5.12, la décroissance de l'erreur RMS en fonction du temps de calcul, entre la méthode incrémentale, l'algorithme GN-MG et la méthode multi-incrémentale.

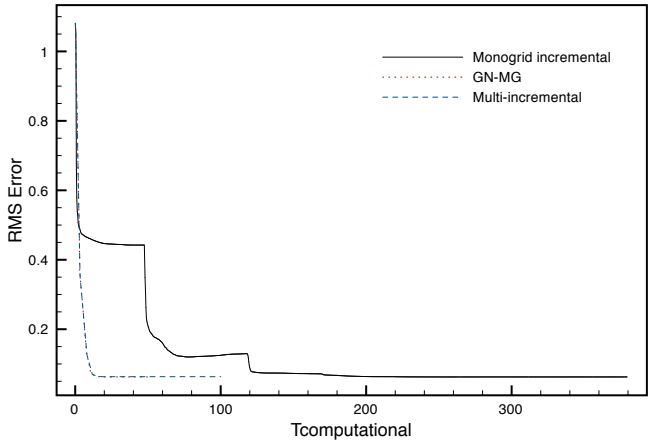

FIG. 5.12 – Décroissance de l'erreur RMS (5.28) pour l'incrémental et le GN-MG et le multi-incrémental, en fonction du temps de calcul en secondes. Pour cette échelle de temps, le GN-MG et le multi-incrémental se confondent.

Selon l'algorithme utilisé, un calcul de l'erreur RMS est fait pour chaque itération de minimisation de la fonction coût linéaire sur la grille fine, ou après correction par grille grossière. L'erreur RMS est aussi calculée lors de la mise à jour du modèle non linéaire à la fin de chaque boucle externe.

Dans les deux cas, FAS et GN-MG, nous vérifions les résultats déjà obtenus dans le cas de l'assimilation linéaire : utiliser une méthode multigrille peut permettre de réduire sensiblement le nombre de calculs sur la grille haute résolution et le coût de calcul sans détériorer le résultat final.

Nous comparons plus en détails, le multi-incrémental et les méthodes multigrilles entre-elles. Figure 5.13 est représentée la décroissance de l'erreur RMS en fonction du temps de calcul pour chaque méthode. Les symboles représentent un calcul de l'erreur RMS, que ce soit avec le modèle linéaire ou avec le modèle non linéaire.

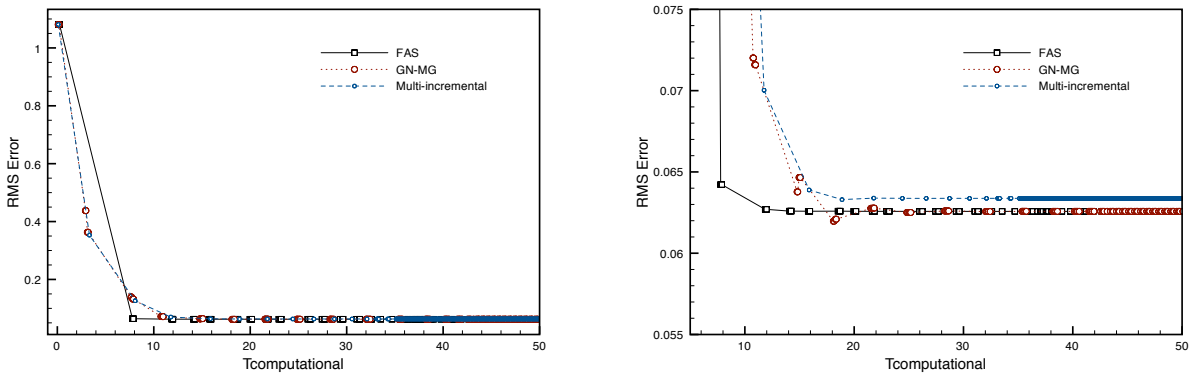

FIG. 5.13 – Décroissance de l'erreur RMS (5.28) en fonction du temps de calcul (s) pour le FAS, le GN-MG et le multi-incrémental. Zoom à droite sur la valeur de convergence.

La vitesse de convergence du multi-incrémental se comporte sensiblement comme le GN-MG. Cependant le multi-incrémental ne converge pas vers la même valeur.

Entre le FAS et le GN-MG, les résultats sont quasiment les mêmes. La raison de préférer le FAS est que cet algorithme repose sur moins de contraintes de convergence que la méthode GN-MG, et nécessite moins de paramètres à définir. Pour cela, le FAS semble plus robuste.

Les symboles de chaque courbe représentent un calcul de l'erreur RMS sur la grille fine.
Dans le cas du multi-incrémental, cela correspond simplement à une mise à jour du misfit. Donc entre chaque symbole, une optimisation basse résolution est effectuée. Cela donne une idée du temps mis à cette résolution. Plus on approche de la convergence, plus ce temps est court.
En ce qui concerne le FAS, nous effectuons des cycles d'une itération sur la grille fine, une résolution sur la grille grossière puis de nouveau une itération sur la grille fine. Cela veut dire que le premier carré noir correspond à la première itération de la grille fine suivie d'un autre symbole correspondant au résultat après la correction par grille grossière. Le premier passage par la grille grossière se termine à peu près à 10 secondes de temps de calcul. La convergence est alors quasiment atteinte. Les autres passages par la grille grossière n'ont alors que peu d'effet sur la minimisation.
Enfin, la méthode GN-MG est composée de V-cycles de 2 itérations sur la grille fine avec entre les deux, un passage par la grille grossière. Le premier symbole correspond donc à la première itération sur la grille fine. Le deuxième symbole, situé entre 2 et 4 secondes correspond à la correction après le passage par la grille grossière. Enfin le troisième symbole, situé juste en dessous représente la mise à jour du modèle non linéaire. Comme dans le cas du FAS, les itérations sur la grille fine sont très rapides et ont peu d'effets. C'est pourquoi le symbole correspondant à la dernière itération sur la grille fine, avant la mise à jour non linéaire est confondu avec les symboles de la correction par la grille grossière. Le GN-MG converge, quant à lui, en à peu près

5 boucles externes, c'est à dire aux alentours de 20 secondes de temps de calcul.

Nous observons que dans notre configuration, un passage par la grille grossière est relativement coûteux, par rapport à une itération sur la grille fine, en particulier lors des premières corrections par grille grossière. Le nombre d'itérations nécessaires pour atteindre le critère de convergence de M1QN3 sur la grille grossière est de l'ordre de 1000 pour les premiers cycles de FAS puis diminue peu à peu pour atteindre une itération lorsqu'on approche la convergence globale. Même si le modèle direct de la grille grossière est 4 fois moins coûteux, les premières 1000 itérations de minimisation sur la grille grossière demandent plus de temps de calcul que les quelques itérations de la grille fine. Cependant, les multigrilles convergent dix fois plus vite que les monogrilles. Et surtout, si l'on passe à un modèle à deux dimensions d'espace avec un facteur de raffinement de 3 en temps et en espace, comme cela va être le cas dans le chapitre suivant, nous obtenons alors un calcul sur la grille grossière 27 fois moins coûteux.
Nous remarquons également que la résolution sur la grille grossière prend légèrement plus de temps pour le FAS que pour le GN-MG. Lors des premiers cycles, le GN-MG est la plus rapide des méthodes à converger sur la grille grossière, car le critère de convergence est atteint pour 700 itérations. Cependant la diminution, induite par un passage sur la grille grossière, est moins performante.

5.3.5 Préconditionnement

Nous avons appliqué le préconditionnement par $\mathbf{B}^{\frac{1}{2}}$ au système d'optimisation. La fonction coût étant similaire à celle du cas test linéaire (voir section 4.3.3), nous avons repris le même principe. La configuration est celle des tests précédents, à un changement de γ près. En effet, avec $\gamma = 0.01$, il s'est avéré impossible de trouver des paramètres de la méthode de minimisation M1QN3, réussissant à diminuer la fonction coût préconditionnée ; alors qu'aucun problème ne survenait pour la fonction non préconditionnée (voir résultats précédents). Après augmentation progressive de γ, il se trouve qu'à partir de la valeur 0.01875, la minimisation devient possible.
Cela rappelle les résultats trouvés en linéaire, section 4.3.3, où selon les valeurs des paramètres, le conditionnement du Hessien peut être augmenté après préconditionnement par $\mathbf{B}$, au lieu d'être diminué. Nous n'avons pas été jusqu'à calculer les valeurs limites des paramètres, pour lesquelles le préconditionnement devient inefficace. Mais nous supposons que c'est pour cette raison que M1QN3 diverge pour $\gamma < 0.01875$.

Pour la valeur de $\gamma = 0.01875$, nous avons testé le monogrille non quadratique avec et sans préconditionnement, et le multigrille non quadratique FAS, avec et sans préconditionnement également. Sur la Figure 5.14, nous pouvons observer la décroissance de l'erreur RMS (5.28) en fonction du temps de calcul, pour les 4 méthodes à gauche et pour les méthodes préconditionnées à droite. Pour les différencier, les méthodes préconditionnées sont précédées de "PC" devant leur nom usuel.

Que ce soit avec ou sans préconditionnement, la méthode multigrille est constituée de cycles comprenant une itération sur la grille fine, suivie d'un passage sur la grille grossière, puis d'une itération sur la grille fine. Elle a la particularité de commencer par un passage par la grille grossière, avant d'effectuer la série de cycles. Chaque triangle représente un calcul sur la grille fine. Le premier correspondant au calcul de l'erreur initiale, le second signifie le résultat après le passage par la grille grossière. Ensuite, il y a alternance, entre une itération sur la grille fine et un passage

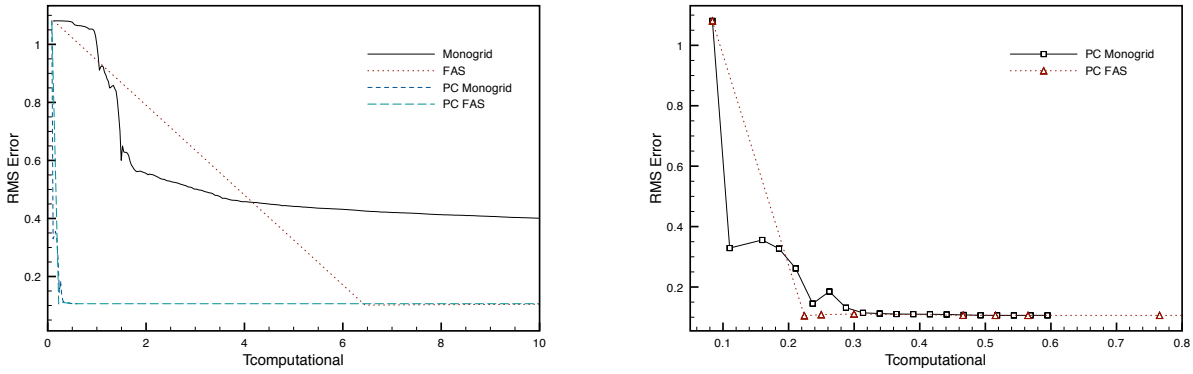

FIG. 5.14 – Décroissance de l'erreur RMS (5.28) en fonction du temps de calcul (s) pour la méthode monogrille et le FAS avec et sans préconditionnement, sur la gauche et avec préconditionnement uniquement sur la droite.

sur la grille grossière. Les deux méthodes FAS convergent quasiment après le premier passage sur la grille grossière ; à 6 secondes de temps de calcul dans le cas non préconditionné, et à 0.2 secondes dans le cas préconditionné.
Dans le cas du monogrille, un carré représente simplement une itération sur la grille fine. Le préconditionnement se révèle ici particulièrement efficace puisqu'il réduit le temps de convergence de 200 à 0.3 secondes.

La méthode multigrille sans préconditionnement a l'avantage de ne pas avoir à inverser $\mathbf{B}^{\frac{1}{2}}$, cependant le préconditionnement demeure plus efficace à accélerer la convergence. Nous remarquons également que la méthode multigrille est encore efficace sur la fonction coût préconditionnée.
Cependant, nous constatons que, pour certaines valeurs de γ (par exemple, $\gamma = 0.01$), le préconditionnement n'est pas possible, c'est à dire qu'il est impossible de trouver de bons paramètres à M1QN3 sans qu'il n'y ait divergence ; alors que le multigrille est efficace. Pour être certain que ce n'est pas dû à la méthode d'optimisation, il faudrait essayer avec d'autres solveurs adaptés aux non linéarités. Nous pouvons néanmoins conclure que les méthodes multigrilles ont un impact considérable, sur une classe plus large de fonctions coût que le préconditionnement, pour la méthode d'optimisation utilisée. De plus, sur des cas tests plus complexes et de plus grande taille, le préconditionnement s'avère coûteux et difficile à mettre en œuvre, ce qui peut avantager l'emploi des méthodes multigrilles.

5.4 Conclusion

Nous venons de présenter l'application des méthodes multigrilles appliquées à l'assimilation de données, dans le cas d'un modèle non linéaire. Nous avons introduit ici deux algorithmes multigrilles, le FAS non quadratique et le GN-MG, qui est appliqué à la méthode incrémentale. Les deux algorithmes ont des conditions similaires de convergence du passage par la grille grossière. Nous nous sommes également intéressés à la méthode multi-incrémentale qui peut être vue comme une variante du GN-MG, pour laquelle aucune minimisation sur la grille fine n'est réalisée.

Nous avons repris la forme de la fonction coût étudiée dans le cas linéaire. Cette forme implique l'ellipticité du Hessien et permet un préconditionnement assez simple à mettre en œuvre.

Sans faire d'étude théorique supplémentaire, nous avons testé les différents algorithmes sur la minimisation de la fonction coût dans le cas d'une équation de Burgers créant un choc aux alentours de $t = 0.2s$.
Dans un premier temps, nous nous sommes intéressés à la validité de certaines hypothèses. Les résultats de l'algorithme incrémental et du GN-MG nous ont ainsi permis de calculer les erreurs de linéarisation et de conclure quant à la plus forte sensibilité du GN-MG aux non-linéarités. Cet algorithme a ainsi plus de difficultés à converger et l'hypothèse de linéarisation ne semble pas toujours vérifiée.
Les erreurs d'approximation de la grille grossière ont été évaluées pour quelques corrections, le comportement de ces erreurs est plus acceptable pour le FAS que pour le GN-MG. En effet, le choc en $t = 0.2$ a un impact marqué sur les erreurs du GN-MG qui augmentent fortement et dans certains cas continuent à augmenter après le choc ; si bien que cela peut conduire à la divergence de la méthode GN-MG au sein d'une boucle externe. Dans tous les cas, avant le choc ($t < 0,3s$), la pente des erreurs d'approximation est plus petite que celle induite par les erreurs de linéarisation. Ce qui laisse à penser que la propriété d'approximation est mieux respectée que l'hypothèse de linéarité, pour le cas test décrit.
Nous avons ensuite testé différentes configurations d'algorithmes en variant la valeur de certains paramètres. Nous nous sommes intéressés au nombre d'itérations sur la grille fine, sur la grille grossière et au nombre de cycles par boucle externe dans le cas de l'algorithme GN-MG. Les expériences nous ont permis de conclure que le nombre d'itérations sur la grille fine ainsi que celui sur la grille grossière ne sont pas des paramètres très sensibles. A l'inverse, les difficultés de convergence du GN-MG peuvent être neutralisées en imposant un nombre de cycles par boucle externe égal à un.
Suite à ces expériences, nous avons fixé pour chaque algorithme des cycles d'une itération sur la grille fine, puis convergence de la minimisation sur la grille grossière, suivie d'une dernière itération sur la grille fine. Ces paramètres nous ont permis de comparer les différents algorithmes entre eux. Le FAS, comme le GN-MG, permet de réduire d'un facteur 10 le temps de convergence du monogrille associé. Les deux algorithmes multigrilles, ainsi que le multi-incrémental, ont un comportement relativement similaire. Cependant ce dernier ne converge pas vers la même solution. Nous notons qu'un passage par la grille grossière reste relativement coûteux (nous allons à convergence) mais il est beaucoup plus efficace que de minimiser tout le temps sur la grille fine. De plus, avec un modèle à deux ou trois dimensions et un facteur de raffinement plus grand, le coût de la correction grille grossière peut être considérablement réduit.
En somme, l'algorithme GN-MG semble moins robuste et comporte plus de paramètres à fixer ; ce qui nous fait préférer le FAS.

Nous concluons le chapitre en étudiant l'influence du préconditionnement. Celui-ci est très efficace et permet la convergence en une dizaine d'itérations du système non linéaire. Le temps de calcul se trouve alors réduit de 200 à 0.3 secondes. Cependant la minimisation de la fonction coût préconditionnée peut être encore accélérée par une méthode multigrille.
Nous remarquons également l'impossibilité de préconditionner par $\mathbf{B}$ certaines fonctions selon les valeurs des paramètres utilisées. Ces restrictions ne concernent apparemment pas les méthodes multigrilles qui peuvent être appliquées sur une gamme plus large de fonctions coûts.

Les méthodes multigrilles ont montré leur efficacité sur un modèle non linéaire relativement simple. Nous nous intéressons dans le chapitre suivant à des modèles plus complexes aux dimensions beaucoup plus élevées.

6

APPLICATION À UN MODÈLE SHALLOW WATER

Sommaire

6.1 Introduction . . . 135
6.2 Variational data assimilation . . . 136
6.2.1 A complex control optimal system . . . 136
6.2.2 Incremental methods to solve the optimal system . . . 136
6.3 Multigrid methods . . . 138
6.3.1 A way to accelerate iterative methods . . . 138
6.3.2 Convergence properties . . . 139
6.4 The Multigrids applied to 4D-Var . . . 140
6.4.1 Non linear minimization . . . 140
6.4.2 Gauss-Newton Multigrid . . . 141
6.4.3 Convergence properties . . . 141
6.5 A 2D Shallow water model . . . 142
6.5.1 The Shallow water equations . . . 143
6.5.2 The multigrid configuration . . . 144
6.5.3 Experimental set-up . . . 144
6.5.4 Numerical results without preconditionning . . . 147
6.5.5 Preconditionned cost function . . . 150
6.6 Conclusion . . . 152

Résumé en Français

Ce chapitre présente les résultats obtenus en implémentant les algorithmes multigrilles présentés au chapitre précédent sur un modèle Shallow water à deux dimensions d'espace. Il est présenté sous forme d'article que nous allons soumettre.
Pour cette raison, les premières sections reprennent des résultats déjà énoncés. Il est à noter cependant que l'on introduit, pour l'algorithme FAS, une recherche linéaire à la section 6.4.3.
Dans la section 6.5, nous présentons le modèle Shallow water, les différentes résolutions, le processus d'assimilation ainsi que les résultats pour deux fonctions coûts différentes. En effet, nous avons résolu deux problèmes, le premier non préconditionné, le deuxième avec préconditionnement.
Dans le cas du système préconditionné, l'utilisation de la relation de Galerkin pour définir la matrice **B** sur la grille grossière a été nécessaire pour obtenir de bons résultats.

Les résultats sont satisfaisants, que ce soit avec le FAS ou le GN-MG, et avec ou sans préconditionnement ; même si ils mettent en avant la nécessité d'intégrer de manière plus complexe les corrections par grille grossière à l'algorithme d'optimisation.

The use of multigrid methods to solve a variational data assimilation system : Experiments on a 2D Shallow water model.

Emilie Neveu, Laurent Debreu, François-Xavier LeDimet

Abstract :
In order to limit the computational cost of the variational data assimilation process, we study the application of the multigrid methods to solve this optimal control system. These methods are used to accelerate an iterative method by solving for the error on low resolution grids. We define two multigrid algorithms and their convergence properties. The first one is based on the Full Approximation Scheme (FAS), a non-linear multigrid algorithm and the second one is the Gauss-Newton Multigrid (GN-MG), the multigrids applied to the incremental variational method, a Gauss-Newton method. We use a 2D Shallow-water model to compare them to the algorithms commonly used on operational context, named incremental and multi-incremental methods. Particularly, we emphasize the necessity of using a linesearch to ensure the convergence of the algorithm. The algorithms are also compared on a preconditionned optimal control system, for which the multigrids are still more efficient than the monogrid methods.

6.1 Introduction

Data assimilation methods are a way of combining different sources of information : observations and numerical models according to error statistics on these sources. They can be divided into two groups [3]. First, sequential methods are based on the Kalman filtering approach and make the state vector evolve in time along with its error statistics. Then, variational methods are based on optimal control techniques and minimize the distance between the model trajectory and observations according to a cost function $\mathcal{J}$. These methods have lead to strong improvements in the operational context of weather or ocean forecast. But they still have huge computational costs and have to be simplified for operational purposes. We will focus on the 4D-variational data assimilation (4D-var), introduced by LeDimet and Talagrand in 1986 [39]. For non linear models and operational context, simplified algorithms, that deal with minimizing a sequence of quadratic cost functions, have been implemented. They are the incremental (Courtier [11]), and the multi-incremental (Veersé *et al.*[63]) approaches. The latter one is fast, using a coarser resolution to solve the system. But it does not converge to the original solution.

In the optimal control framework, several attempts have been made to apply multigrid methods, either for linear or non linear optimization (Ta'asan [58], Nash [48], [40]) but never on data assimilation systems. In variational data assimilation, assuming $\mathbf{x}_0$ is the control vector, the necessary condition of optimality is given by the Euler-Lagrange equation $\nabla_{\mathbf{x}_0}\mathcal{J} = 0$. In this paper, the multigrid methods are used to solve the resulting system.
We present in section 6.2 the variational data assimilation optimal control system. We introduce the multigrid methods in the general case of non linear systems in section 6.3. These methods are used to accelerate an interative method by solving the error on low resolutions grids. We then adapt the multigrid methods to variational data assimilation in section 6.4, in which it will be introduced two algorithms, the Full Approximation Scheme (FAS), based on non linear

minimization and the Gauss-Newton Multigrid (GN-MG) based on the incremental approach. The multigrid method needs the coarse grid operators to be close to the original fine grid ones in order to converge. With the FAS algorithm, it's also necessary to use a line search for the coarse grid correction step. In section 6.5, we apply the multigrid methods to solve data assimilation optimal system on a 2D Shallow-water equation. We compare the multigrid algorithms with monogrid ones, with and without preconditionning.

6.2 Variational data assimilation

6.2.1 A complex control optimal system

In variational data assimilation method (Le Dimet *et al*, 86 [39], Ide *et al*, 97 [32], Bennett *et al*, 02[3]), the aim is to minimize a given cost function $\mathcal{J}(\mathbf{x}_0)$ playing with the control vector $\mathbf{x}_0$, that can be the initial state of the model. The cost function measures the distance between a set of observations $\mathbf{y}^o$ (available for times t_i) and the model as follows

$$\mathcal{J}(\mathbf{x}_0) = \frac{1}{2}\sum_i \|H_{t_i}(\mathbf{x}_0) - \mathbf{y}^o_{t_i}\|^2_{\mathbf{R}^{-1}_{t_i}} + \frac{1}{2}\|\mathbf{x}_0 - \mathbf{x}^b\|^2_{\mathbf{B}^{-1}} \tag{6.1}$$

The first term measures the misfit to data while the last term is a regularization one, $\mathbf{x}^b$ being the background or current estimate of the initial state.
H_{t_i} is an operator that includes both the model trajectory $M_{0,i}(.)$ from time 0 to time i and the observation operator that goes from the model space to the observations space. $\mathbf{B}$ is the background error covariance matrix associated to $\mathbf{x}^b$ and $\mathbf{R}_{t_i}$ the observational error covariance matrix.

Let $\mathbf{x}^a_0$ be the minimum of the cost function : $\mathcal{J}(\mathbf{x}^a_0) = \min_{\mathbf{x}_0} \mathcal{J}(\mathbf{x}_0)$.
Then, a necessary condition for $\mathbf{x}^a_0$ to be a minimum of $\mathcal{J}(\mathbf{x}_0)$ is the Euler equation :

$$\nabla_{\mathbf{x}_0}\mathcal{J}(\mathbf{x}^a_0) = 0 \tag{6.2}$$

To find the minimum, variational data assimilation uses an optimization procedure, that requires the knowledge of the gradient of $\mathcal{J}$. This gradient can be efficiently computed using the adjoint model (Le Dimet [39]). As one integration of an operational model is already of a high cost of solution, solving data assimilation control optimal system can then become very expensive. If we use the model without reduction of physics or dimension, the data assimilation cost can reach 100 times the model cost (in the case of the incremental approach used in Météo France in the year of 2010). In addition, the non-linearities and the complexities of physical phenomena make the resulting non-quadradic cost function difficult to minimize.

6.2.2 Incremental methods to solve the optimal system

To deal with the non-linearities, the incremental 4D-Var has been proposed (Courtier *et al*, 94 [11]). It is an iterative algorithm based on the Gauss-Newton method.
Assuming non-linearities are small, the model and observation operator are linearized around the kth iterate $\mathbf{x}_{0,k}$, during the period of integration :

$$H_{t_i}(\mathbf{x}_{0,k+1}) - H_{t_i}(\mathbf{x}_{0,k}) \simeq \mathbf{H}_{t_i}\delta\mathbf{x}_{0,k}, \quad \forall i$$

using the tangent linear operator $\mathbf{H}_{t_i} = \left.\dfrac{\partial H_{t_i}}{\partial \mathbf{x}_0}\right|_{\mathbf{x}_{0,k}}$.
The cost function (6.1) can then be written in terms of $\delta\mathbf{x}_{0,k}$:

$$\mathcal{J}[\delta\mathbf{x}_{0,k}] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}_{t_i}\delta\mathbf{x}_{0,k} - \mathbf{d}_{i,k}\|^2_{\mathbf{R}_{t_i}^{-1}} + \frac{1}{2}\|\delta\mathbf{x}_{0,k} - \delta\mathbf{x}_k^b\|^2_{\mathbf{B}^{-1}} \tag{6.3}$$

where $\mathbf{d}_{i,k} = \mathbf{y}_{t_i} - H_{t_i}(\mathbf{x}_{0,k})$ is called the misfit and $\delta\mathbf{x}_k^b = \mathbf{x}^b - \mathbf{x}_{0,k}$.
This new cost function is now a quadratic one, which is much easier to minimize.
After each minimization, the non-linear model is integrated from the new estimate $\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\mathbf{x}_{0,k}$ to update the misfit and $\delta\mathbf{x}_k^b$.

Incremental 4D-Var algorithm :
$\mathbf{x}_{0,k=0} = \mathbf{x}^b$ is the first estimate
Outer-loop on k until convergence

1. $\mathbf{d}_{i,k} = \mathbf{y}_{t_i} - H_{t_i}(\mathbf{x}_{0,k}), \quad \delta\mathbf{x}_k^b = \mathbf{x}^b - \mathbf{x}_{0,k}$
2. *Inner-loop :* Find the minimum $\delta\hat{\mathbf{x}}_{0,k}$ of $\mathcal{J}[\delta\mathbf{x}_{0,k}]$ with

$$\mathcal{J}[\delta\mathbf{x}_{0,k}] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}_{t_i}\delta\mathbf{x}_{0,k} - \mathbf{d}_{i,k}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\delta\mathbf{x}_{0,k} - \delta\mathbf{x}_k^b\|^2_{\mathbf{B}^{-1}}$$

3. Correct the iterate,

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \delta\hat{\mathbf{x}}_{0,k} \tag{6.4}$$

4. $k = k + 1$

End of outer-loop

To deal with the complexities of physical phenomena or the huge scale of a discretized problem, the linearized model can also be simplified to a so-called reduced model by using $\delta\mathbf{x}_0^s = S(\delta\mathbf{x}_0)$, S being an operator of simplification, $\mathbf{S}$ its tangent linear and $\mathbf{S}^{-I}$ its pseudo-inverse. The new algorithm is then :

Reduced Incremental 4D-Var algorithm :
$\mathbf{x}_{0,k=0} = \mathbf{x}^b$ is the first estimate
Outer-loop on k until convergence

1. $\mathbf{d}_{i,k} = \mathbf{y}_{t_i} - H_{t_i}(\mathbf{x}_{0,k}), \quad (\delta\mathbf{x}_k^b)^s = S(\mathbf{x}^b - \mathbf{x}_{0,k})$
2. *Inner-loop :* Find the minimum $\delta\hat{\mathbf{x}}_{0,k}^s$ of $\mathcal{J}[\delta\mathbf{x}_{0,k}^s]$ with :

$$\mathcal{J}[\delta\mathbf{x}_{0,k}^s] = \frac{1}{2}\sum_{i=0}^{N-1} \|\mathbf{H}_{t_i}^s\delta\mathbf{x}_{0,k}^s - \mathbf{d}_{i,k}\|^2_{\mathbf{R}^{-1}} + \frac{1}{2}\|\delta\mathbf{x}_{0,k}^s - (\delta\mathbf{x}_k^b)^s\|^2_{(\mathbf{B}^s)^{-1}} \tag{6.5}$$

3. Correct the iterate,

$$\mathbf{x}_{0,k+1} = \mathbf{x}_{0,k} + \mathbf{S}^{-I}\delta\hat{\mathbf{x}}_{0,k}^s. \tag{6.6}$$

4. $k = k + 1$

End of outer-loop

$\mathbf{H}_{t_i}^s$, introduced in the cost function (6.5), includes now the reduced model trajectory $M_{0,i}^s(.)$ from time 0 to time i and the observation operator that goes from the reduced model space to

the observation space. $\mathbf{B}^s$ is supposed to be equal to $\mathbf{B}^s = \mathbf{SBS}^T$, it represents the reduced background error covariance matrix.
This reduced algorithm gave rise to the multi-incremental algorithm(Veersé *et al.* [63]), by replacing S with an operator of reduction that goes from the original grid to a coarser grid.

Convergence properties of these algorithms are found in Gratton *et al.* [23]. The incremental algorithm converges to $\mathbf{x}_0^a$ for $\mathbf{x}^b$ close enough to $\mathbf{x}_0^a$, if the tangent linear hypothesis is verified. On the contrary the reduced incremental algorithm converges to $\mathbf{x}_0^{a'}$ for $\mathbf{x}^b$ close enough to $\mathbf{x}_0^{a'}$, if the tangent linear hypothesis on $\mathbf{H}_i^s$ is true. The minimum $\mathbf{x}_0^{a'}$ is different from $\mathbf{x}_0^a$, the distance between both depends on the distance between $\mathbf{H}_i$ and $\mathbf{H}_i^s$.

6.3 Multigrid methods

Multigrid methods are known since years 70 (Southwell 1935 [56], 1946 [57]) and have been used for a large class of systems, from linear elliptic ones to Navier Stokes equations (Brandt, 1980 [6],[7]). In this section, basic ideas are reminded. The next section will discuss different ways of using a multigrid method for the data assimilation problem.

6.3.1 A way to accelerate iterative methods

Let $\mathbf{x}_0^a$ be the solution of the following, potentially non linear, system :

$$A(\mathbf{x}_0^a) = \mathbf{g} \quad \text{on } \Omega \tag{6.7}$$

Let $\mathbf{x}_0$ be an approximation of $\mathbf{x}_0^a$ and $\delta\mathbf{x}_0 = \mathbf{x}_0^a - \mathbf{x}_0$ the error.
The residual $\mathbf{r}$, given by $\mathbf{r} = \mathbf{g} - A(\mathbf{x}_0)$, satisfies the residual equation

$$A(\mathbf{x}_0 + \delta\mathbf{x}_0) - A(\mathbf{x}_0) = \mathbf{r} \tag{6.8}$$

Commonly, iterative methods are used to solve equation (6.7). For elliptic systems, it exists a whole-range of iterative methods called *smoothers* that reduce efficiently small scales but have difficulties in reducing large scales of the error $\delta\mathbf{x}_0$. Gauss-Seidel is one of them. Multigrid methods use different levels of grid resolution to improve these iterative methods, solving more efficiently the residual equation (6.8). Indeed the remaining large scales, when described by a coarser grid resolution, are seen as higher scales.
In the following, for sake of clarity, we restrict ourselves to a two-level multigrid method. We assume we solve the system (6.7) on a fine grid domain Ω^f. The superscript f (resp. c) stands for the fine (resp. coarse) grid. To exchange informations between the two grids, we use restriction operators I_f^c, $\hat{I}_f^c$ and an interpolation or prolongation operator I_c^f. $\hat{I}_f^c$ is the restriction operator used to transpose $\mathbf{x}_0^f$. It may be different from the one transposing the residual $\mathbf{r}^f$, because $\mathbf{x}_0^f$ and $\mathbf{r}^f$ does not live on the same space. We denote by A^c, A^f and $\mathbf{g}^c$ and $\mathbf{g}^f$ the discretizations of A and $\mathbf{g}$ on the coarse and fine grids.
The algorithm called the Full Approximation Scheme (FAS) can be introduced :

Full Approximation Scheme algorithm :
Loop on k until convergence

1. Apply ν_1 times an iterative method named SMOOTH :

$$\mathbf{x}_0^f = SMOOTH^{\nu_1}(\mathbf{x}_{0,k}^f, A^f, \mathbf{g}^f) \tag{6.9}$$

2. Coarse grid correction step :
   (a) Solve exactly
$$A^c(\mathbf{x}_0^c) - A^c(\hat{I}_f^c \mathbf{x}_0^f) = r^c = I_f^c(\mathbf{g}^f - A^f(\mathbf{x}_0^f)) \tag{6.10}$$
   where $\mathbf{r}^c$ is the fine grid residual transfered to the coarse grid : $\mathbf{r}^c = I_f^c \mathbf{r}^f$.
   (b) Interpolate the correction and deduce the approximate solution on Ω^f, which may be expressed by :
$$\mathbf{x}_{0,\text{ after CGC}}^f = \mathbf{x}_0^f + I_c^f(\mathbf{x}_0^c - \hat{I}_f^c \mathbf{x}_0^f) \tag{6.11}$$
3. Apply ν_2 times the iterative method :
$$\mathbf{x}_{0,k+1}^f = SMOOTH^{\nu_2}(\mathbf{x}_{0,\text{ after CGC}}^f, A^f, \mathbf{g}^f) \tag{6.12}$$

End Loop, $k = k + 1$

SMOOTH is an iterative method suited for non linear systems. The residual equation (6.10) can be solved recursively by a multigrid method if more than 2 grids are used. Then the algorithm is recursive.
In the following, the residual equation (6.10) will be written as :
$$A_{\text{FAS}}^c(\mathbf{x}_0^c) = I_f^c \mathbf{g}^f \tag{6.13}$$
with
$$A_{\text{FAS}}^c(\mathbf{x}_0^c) = A^c(\mathbf{x}_0^c) + I_f^c A^f(\mathbf{x}_0^f) - A^c(\hat{I}_f^c \mathbf{x}_0^f) \tag{6.14}$$

6.3.2 Convergence properties

The algorithm converges to $\mathbf{x}_0^{f,a}$ depending on the operators and relaxation method used. First, for linear case where $A = \mathbf{A}$; if the operator A is elliptic, it's possible to find iterative methods that, if applied on the fine grid, will efficiently remove the small scales in the error $\delta\mathbf{x}_0^f$ (see Trottenberg [62]). For non linear cases, this convergence analysis is not possible. However if the ellipticy insures that the multigrids will be more efficient than monogrid, it's not a necessary condition for convergence. Even if it is non linear or with small ellipticity, the multigrid methods may still be efficient.
The necessary condition for convergence is the good behavior of the coarse grid correction step. The coarse grid model should be consistent with the fine grid model. One way to be sure of the quality of the approximation is to approximate the Galerkin relation (Trottenberg [62]) :
$$\|A^c - I_f^c A^f I_c^f\| \leq (\Delta x^f)^\alpha, \quad \alpha > 0 \tag{6.15}$$
where α is typically at least equal to the order of the consistency of the discretizations.
The first possibility to construct A^c is to use finite difference schemes with larger time and/or space discretization steps as fine grid operators. A second one is to define it according to the Galerkin relation $A^c = I_f^c A^f I_c^f$, commonly used for finite element discretizations (Hemker [28], McCormick [43]).

The multi-incremental algorithm [63], introduced earlier, can be seen as a simplified "multigrid" method that does not apply minimization steps at high resolution. Because it solves an approximation of the optimal control system, it doesn't converge to the true solution $\mathbf{x}_0^{f,a}$.

6.4 The Multigrids applied to 4D-Var

Our objective is to alleviate the cost of 4D-Var calculation by solving the Euler equation (6.2) using a multigrid method. Nash [48], and Ta'asan [59] have studied some optimal control problems solved by multigrid methods. There, we look more specifically at optimal control problems that come from variational data assimilation problems. They differ by the shape of the cost function.
In order to relate this problem to the one introduced in the previous section, we write the Euler equation (6.2) under the form $A(\mathbf{x}_0) = g$ where $\nabla J(\mathbf{x}_0) = A(\mathbf{x}_0) - g$.

This is the direct application of the FAS. The alternative named Gauss-Newton Multigrid, or GN-MG, uses the FAS algorithm to solve the *inner-loop* minimizations of the incremental method.

6.4.1 Non linear minimization

The Full Approximation Scheme consists in applying the multigrid algorithm to solve the non linear system $\nabla \mathcal{J}^f(\mathbf{x}_0^f) = 0$. The residual equation solved on coarse grid is inspired by (6.13) and (6.14) :

$$\nabla \mathcal{J}^c_{\text{FAS}}(\mathbf{x}_0^c) = \nabla \mathcal{J}^c(\mathbf{x}_0^c) + I_f^c \nabla \mathcal{J}^f(\mathbf{x}_0^f) - \nabla \mathcal{J}^c(\hat{I}_f^c \mathbf{x}_0^f) = 0 \tag{6.16}$$

By defining

$$L = I_f^c \nabla \mathcal{J}^f(\mathbf{x}_0^f) - \nabla \mathcal{J}^c(\hat{I}_f^c \mathbf{x}_0^f) \tag{6.17}$$

we have to minimize on coarse grid the new cost function $\mathcal{J}^c_{\text{FAS}}$:

$$\mathcal{J}^c_{\text{FAS}}(\mathbf{x}_0^c) = \mathcal{J}^c(\mathbf{x}_0^c) + < L, \mathbf{x}_0^c > \tag{6.18}$$

The iterate $\mathbf{x}_0^f$ is transposed on coarse grid with the restriction operator $\hat{I}_f^c$, supposedly different from I_f^c. Next, we will simply use the same restriction operator for both.
Figure 6.1 shows the schematic of this algorithm.

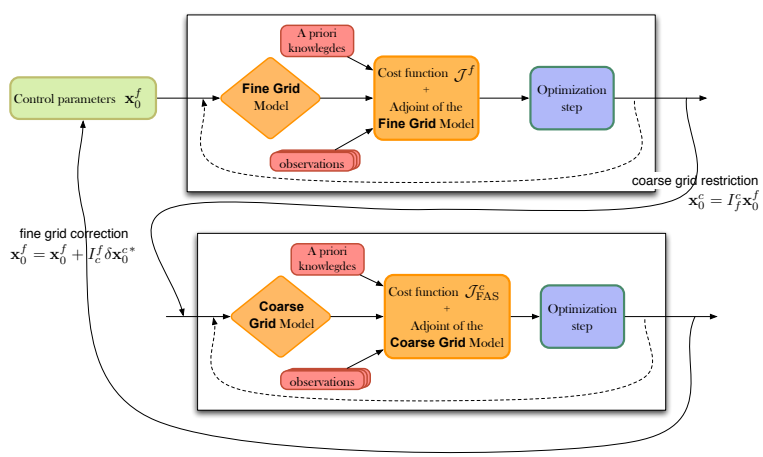

FIG. 6.1 – The 4D-Var FAS Algorithm

6.4.2 Gauss-Newton Multigrid

The Gauss-Newton Multigrid method , or GN-MG, consists in solving the minimization of each quadratic function (6.3) using a multigrid algorithm. Because, the cost function $\mathcal{J}^c(\delta\mathbf{x}_0^c)$ is quadratic, the residual equation (6.16) is then :

$$\mathcal{H}^c(\delta\mathbf{x}_0^c - I_f^c\delta\mathbf{x}_0^f) = -I_f^c\nabla\mathcal{J}^f(\delta\mathbf{x}_0^f) \tag{6.19}$$

with $\mathcal{H} = \nabla^2\mathcal{J}$ the linear Hessian of $\mathcal{J}$.
Also the correction step can be written :

$$\delta\mathbf{x}_{0,\text{afterCGC}}^f = \delta\mathbf{x}_0^f + I_c^f(\delta\mathbf{x}_0^c - I_f^c\delta\mathbf{x}_0^f) \tag{6.20}$$

Using the residual equation (6.19), we can express the coarse grid correction :

$$\mathbf{x}_{0,\,\text{afterCGC}}^f = \mathbf{x}_0^f - \left(I_c^f(\mathcal{H}^c)^{-1}I_f^c\nabla\mathcal{J}^f\right)(\mathbf{x}_0^f) \tag{6.21}$$

when going until convergence on coarse grid.
The coarse grid correction step behaves like an iteration of a Quasi-Newton method that approximates $(\mathcal{H}^f)^{-1}$ with $I_c^f(\mathcal{H}^c)^{-1}I_f^c$.

Figure 6.2 represents the algorithm of GN-MG.

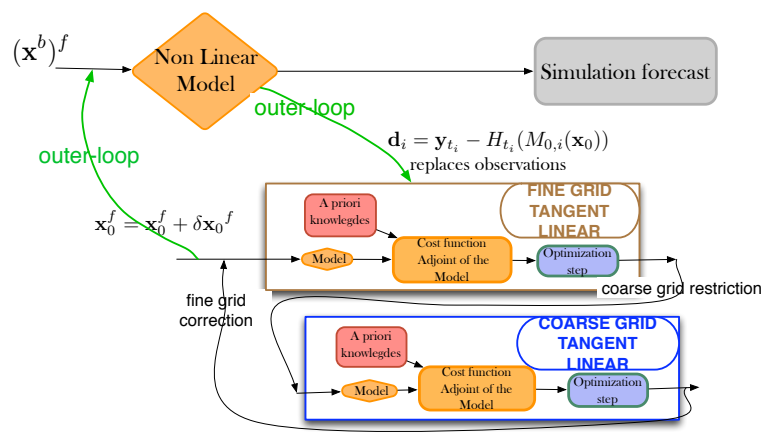

FIG. 6.2 – The 4D-VAR Gauss-Newton Multigrid Algorithm.

The multi-incremental algorithm can be seen as a variant of the GN-MG, for which none fine grid iteration is computed.

6.4.3 Convergence properties

We adapt the convergence properties discussed in section 6.3.2 to the case of a 4D-Var optimal system.

Ellipticity

The **B** matrix is usually used to make the system well-posed and approximates the background covariance error as a Gaussian. It also adds ellipticity to the Hessian.

Coarse grid correction step

The coarse grid correction step, as in equation (6.15), gives us a convergence condition to be applied to the Hessian $\mathcal{H}$:

$$\|\mathcal{H}^c - I_f^c \mathcal{H}^f I_c^f\| \leq (\Delta x^f)^\alpha \tag{6.22}$$

We can express the Hessian using the cost function (6.1) :

$$\mathcal{H} = \sum_i (\mathbf{H}_{t_i})^T \ \mathbf{R}^{-1}\mathbf{H}_{t_i} + \mathbf{B}^{-1} + \sum_i \left(\frac{\partial^2 H_{t_i}}{\partial \mathbf{x}_0^2}\right)^T \ \mathbf{R}^{-1}[H_{t_i}(\mathbf{x}_0) - \mathbf{y}_{t_i}] \tag{6.23}$$

We remark that to satisfy (6.22), we can define the coarse grid operators $\mathbf{B}^c$, $\mathbf{R}^c$and $H_{t_i}^c$ according to the Galerkin relation :

$$H_{t_i}^c = H_{t_i}^f I_c^f, \quad (\mathbf{B}^c)^{-1} = I_f^c (\mathbf{B}^f)^{-1} I_c^f, \quad (\mathbf{R}^c)^{-1} = \frac{1}{c}(\mathbf{R}^f)^{-1}$$

using the fact that interpolation and restriction operators are adjoints so that $I_f^c = c(I_c^f)^T$. Nevertheless we cannot always use these relations, especially the first one which would require an interpolation of the fine grid model. But this may be a good option to try to approach them, for example when defining the coarse grid background error covariance matrix.
In a more general case, we will require that :

$$\begin{cases} \|H_{t_i}^c - H_{t_i}^f I_c^f\| \leq (\Delta x^f)^\alpha, \quad \forall i \\ \|(\mathbf{R}^c)^{-1} - \frac{1}{c}(\mathbf{R}^f)^{-1}\| \leq (\Delta x^f)^\alpha \\ \|(\mathbf{B}^c)^{-1} - I_f^c (\mathbf{B}^f)^{-1} I_c^f\| \leq (\Delta x^f)^\alpha \end{cases} \tag{6.24}$$

Tangent linear hypothesis

The GN-MG algorithm combines the convergence properties of the multigrid methods and of the incremental approach. The validity of the tangent linear hypothesis is function of the degree of non-linearity (and so the model resolution) and of the length of the data assimilation window.

Linesearch

For non-linear problems, relation (6.21) is only a first order approximation. The linesearch is a way to avoid the divergence of the multigrid methods by adding a coefficient α that weights the coarse grid correction (see Nash [40]) :

$$\mathbf{x}_{0,\,\text{afterCGC}}^f = \mathbf{x}_0^f + \alpha I_c^f \delta \mathbf{x}_0^c$$

Note that the linesearch used here follows the Goldstein rules and is a simple algorithm that does not require the computation of the gradient (that would be too costly).

6.5 A 2D Shallow water model

We apply the 4D-Var multigrid algorithms presented above on a 2D Shallow water model. In this section, we introduce first the equations and the grid configuration. We also describe the cost functions used, without and with preconditionning, as well as the data assimilation system

chosen.
We will then implement monogrid high resolution methods and compare them to multigrid methods : the FAS and the GN-MG algorithms. We emphasize the necessity of customizing the optimization method so as to get even better results with the multigrid algorithms.

6.5.1 The Shallow water equations

The Shallow-water equations integrates the water height $\mathbf{h}$, and the horizontal velocity $(\mathbf{u}, \mathbf{v})$.

$$\begin{cases} \dfrac{\partial \mathbf{h}}{\partial t} + \dfrac{\partial(\mathbf{hu})}{\partial x} + \dfrac{\partial(\mathbf{hv})}{\partial y} = 0 \\ \dfrac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\dfrac{\partial \mathbf{u}}{\partial x} + \mathbf{v}\dfrac{\partial \mathbf{u}}{\partial y} - f\mathbf{v} + g\dfrac{\partial \mathbf{h}}{\partial x} + r\mathbf{u} - \nu\Delta\mathbf{u} = \dfrac{\tau_x}{\rho_0 \mathbf{h}} \\ \dfrac{\partial \mathbf{v}}{\partial t} + \mathbf{u}\dfrac{\partial \mathbf{v}}{\partial x} + \mathbf{v}\dfrac{\partial \mathbf{v}}{\partial y} + f\mathbf{u} + g\dfrac{\partial \mathbf{h}}{\partial y} + r\mathbf{v} - \nu\Delta\mathbf{v} = \dfrac{\tau_y}{\rho_0 \mathbf{h}} \end{cases} \tag{6.25}$$

where f is the Coriolis parameter, g the gravity, ν the viscosity coefficient, and r the bottom friction. The evolution of the ocean state is controled by the surface wind stress, $\tau = (\tau_x, \tau_y)$. The domain is closed and no slip boundary conditions ($\mathbf{u} = \mathbf{v} = 0$) are applied.

These equations are integrated on a box-domain $[0, L_x] \times [y_0 - \frac{L_y}{2}, y_0 + \frac{L_y}{2}]$ where y_0 is the value of y on the box center. The Coriolis effect is modelized using β-plane approximation : $f(y) = f(y_0) + \beta(y - y_0)$ with $\beta = \frac{\partial f}{\partial y}(y_0)$. The wind stress is chosen to be a time constant :

$$\tau = \tau_0 \cos\left(\frac{2\pi(y - y_0)}{L_y}\right)\mathbf{i}$$

This model configuration is well-known to model a double-gyre circulation.

The numerical values of the model parameters are given by $L_x = L_y = 2000km, f(y_0) = 0.7 \times 10^{-4}s^{-1}, \beta = 2 \times 10^{-11}m^{-1}s^{-1}, r = 1.10^{-7}m^2s^{-1}, \rho_0 = 1000kg.m^{-3}, g = 0.02m.s^{-2}$, $\tau_0 = 0.015N.m^{-2}$ and $\nu = 27m^2.s^{-1}$, using the same values as Simon [55] and Vidard [66]. The model is discretized using centered finite difference scheme and an Euler forward backward scheme in time. The Arakawa C- grid is used, see Figure 6.3, with the same space step $\Delta x = \Delta y$ in the two directions. More details about the scheme can be found on [54].

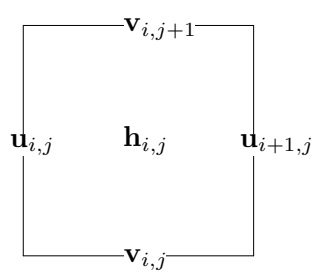

FIG. 6.3 – Arakawa C-grid

6.5.2 The multigrid configuration

We use a 2-grids configuration with a mesh refinement factor equal to 3 in space and time. That makes a coarse-grid model 27-times faster to solve than the fine grid one.

	Grid size	Space step (km)	Time step (s)
High resolution model Ω^f	240×240	$\Delta x^f = \Delta y^f = 25/3$	$\Delta t^f = 24$
Low resolution model Ω^c	80×80	$\Delta x^c = \Delta y^c = 25$	$\Delta t^c = 72$

TAB. 6.1 – 2-grid configuration.

In multigrids, only the initial state is transfered from one grid to another. To do so, we use linear interpolation, and full-weighting restriction.

$$I_f^c = \frac{1}{4}\begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix}_f \qquad I_c^f = \frac{1}{4}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}_c$$

6.5.3 Experimental set-up

We set the initial state $\mathbf{x}_0 = (\mathbf{h}_0, \mathbf{u}_0, \mathbf{v}_0)$ to be the estimated parameter. We integrate the model from a known initial state ; called the true initial state $\mathbf{x}_0^t$ to generate synthetic observations on a finer grid of grid size 720×720. In this test case, we create the true initial state, integrating a model from rest ($\mathbf{h} = \mathbf{h}_0 = 500\,m, \mathbf{u} = \mathbf{v} = 0\,m.s^{-1}$) during 5 years, the spin-up time. This model is calculated on a very high resolution grid ($720 \times 720 : \Delta x = \Delta y = 25/9 \simeq 2.77km$). In Figure 6.4 to the left, we plot the true initial water height.

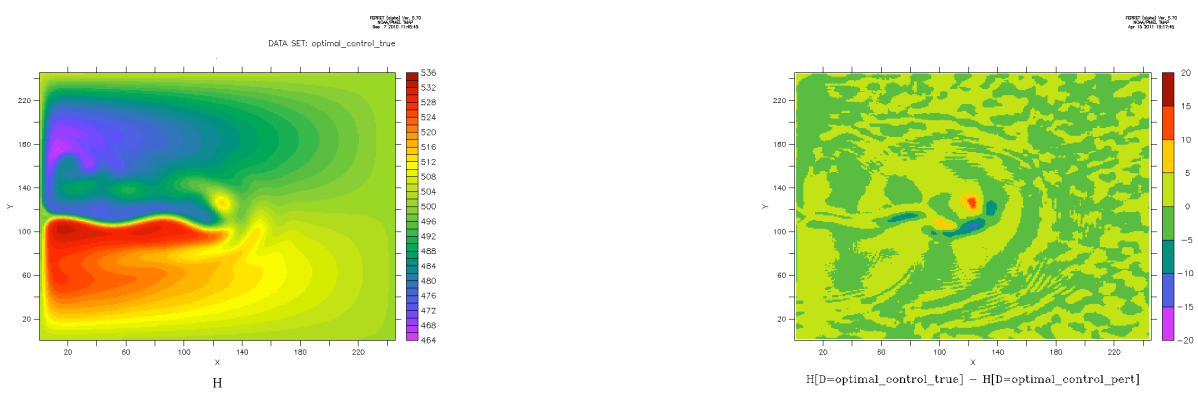

FIG. 6.4 – The initial state for water height $\mathbf{h}$: the true state (left), the true state minus background state(right).

Then we fix a background term, or first guess, named $\mathbf{x}_0^b$, which is a previous analysed state. Indeed it comes from a previous 1-month data assimilation process on the Ω^f domain. This background is created so as to be close enough from the optimal true initial state, because it takes into account the very high resolution data and the high resolution physics modelized by

the fine grid model (see Figure 6.4 he true state minus background state to the right).
To create the data, we integrate the very high resolution model 1 month more. We only observe the water height : $\mathbf{y}^o = \mathbf{h}^t$ and the data are available every 3 days with a sampling in space (on the fine grid, every 12 space steps).
In figure 6.5, we summarize the construction of this data assimilation experimental set-up.

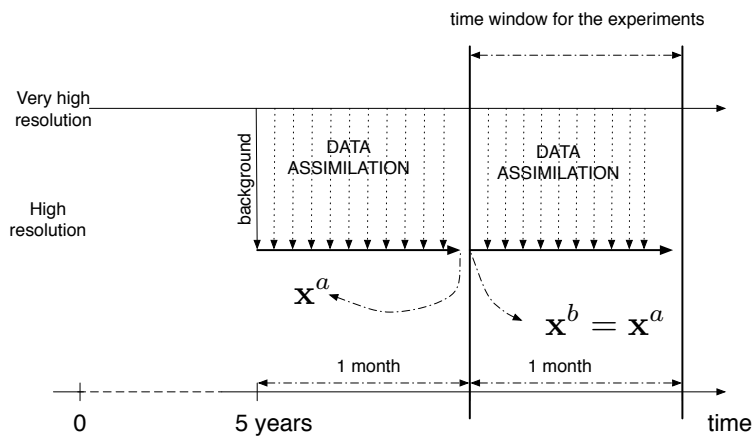

FIG. 6.5 – The high resolution data assimilation process

Cost function $\mathcal{J}$

This data assimilation process is about minimizing the general cost function (6.1) on high resolution grid Ω^f :

$$\mathcal{J}(\mathbf{w}_0) = \frac{1}{2}\sum_i \|H_{t_i}(\mathbf{x}_0) - \mathbf{y}^o_{t_i}\|^2_{R_i^{-1}} + \frac{1}{2}\|\mathbf{w}_0\|^2_{\mathbf{S}^{-1}} \tag{6.26}$$

with

$$\mathbf{x}_0 = \mathbf{K}\mathbf{w}_0 + \mathbf{x}^b, \tag{6.27}$$

$\mathbf{K}$ being a geostrophic balance defined by equation (6.28). This term maintains the initial condition closed to geostrophic balance.

$$\mathbf{K} = \begin{bmatrix} \mathbf{I} & (\mathbf{0}) & (\mathbf{0}) \\ -\frac{g}{f}\frac{\partial}{\partial y} & \mathbf{I} & (\mathbf{0}) \\ \frac{g}{f}\frac{\partial}{\partial x} & (\mathbf{0}) & \mathbf{I} \end{bmatrix} \tag{6.28}$$

The model error matrix covariance R_i is supposed to be a Gaussian with zero mean and a variance of σ_0^2 .
The $\mathbf{S}^{-1}$ matrix is constructed using a smoothing norm that is to say we apply a regularization matrix to $\mathbf{w}_0$:

$$\mathbf{S}^{-1} = \beta\left(\beta_1\mathbf{I} - \beta_2\Delta\right)\mathbf{\Sigma}^{-2}, \qquad (\beta, \beta_1, \beta_2) > 0 \tag{6.29}$$

with

$$\mathbf{\Sigma}^{-1} = \begin{bmatrix} \frac{\mathbf{I}}{\sigma_{b,\mathbf{h}}} & 0 & 0 \\ 0 & \frac{\mathbf{I}}{\sigma_{b,\mathbf{u}}} & 0 \\ 0 & 0 & \frac{\mathbf{I}}{\sigma_{b,\mathbf{v}}} \end{bmatrix} \tag{6.30}$$

and $\sigma^2_{b,\mathbf{h}}$, $\sigma^2_{b,\mathbf{u}}$ and $\sigma^2_{b,\mathbf{v}}$ being the variances.

Preconditionned cost function $\mathcal{J}^p$

To accelerate the convergence of the minimization , it is usually used a preconditionning by $\mathbf{B}$ that gives rise to another cost function, named $\mathcal{J}^p$:

$$\mathcal{J}^p(\mathbf{w}_0^p) = \frac{1}{2}\sum_i \|H_{t_i}(\mathbf{x}_0) - \mathbf{y}^o_{t_i}\|^2_{R_i^{-1}} + \frac{1}{2}\|\mathbf{w}_0^p\|^2 \tag{6.31}$$

with

$$\mathbf{x}_0 = \mathbf{\Sigma}\mathbf{K}\mathbf{C}^{\frac{1}{2}}\mathbf{w}_0^p + \mathbf{x}^b, \tag{6.32}$$

with $\mathbf{K}$ defined by (6.28) and $\mathbf{\Sigma}$ by (6.30).
A most common way [68] is to define directly the background error covariance matrix $\mathbf{B}$ as :

$$\mathbf{B} = \mathbf{\Sigma}\mathbf{K}\mathbf{C}\mathbf{K}^T\mathbf{\Sigma} \tag{6.33}$$

so that we get $\mathbf{B}^{\frac{1}{2}}$ easily :

$$\mathbf{B}^{\frac{1}{2}} = \mathbf{\Sigma}\mathbf{K}\mathbf{C}^{\frac{1}{2}} \tag{6.34}$$

Unfortunately this $\mathbf{B}$ matrix is difficult to invert, that is why, without preconditionning, we use the simpler cost function $\mathcal{J}$ (6.26).
The $\mathbf{C}$ matrix is defined as :

$$\mathbf{C} = \begin{bmatrix} \mathbf{C}_h & 0 & 0 \\ 0 & \mathbf{C}_u & 0 \\ 0 & 0 & \mathbf{C}_v \end{bmatrix} \tag{6.35}$$

For each variable α, $\mathbf{C}_\alpha$ is the spatial correlation matrix. We suppose the spatial correlations to decrease with the distance between both grid points so that $\mathbf{C}_\alpha$ modelizes a Gaussian shape. As proposed by Weaver *et al.* [68], such shape is obtained by solving a numerical diffusion equation. In our test case, this equation solves η as :

$$\frac{\partial \eta}{\partial t} - \kappa\triangle\eta = 0, \quad t \in [0, \tau]$$

with $\triangle$ the 2D Laplacian and κ the diffusion coefficient. The initial state of this equation is the state $\mathbf{w}_0^p$. The coefficient $2\kappa\tau$ approximates the square of the spatial correlation length l of the modelized covariances. To modelize the correlation matrix with a number of M time steps, the coefficient κ is set equal to $\kappa = \dfrac{l^2}{2\tau} = \dfrac{l^2}{2M\Delta t}$. The corresponding discrete operator is noted $\mathbf{L}$.

These modelized covariances have to be normalized to get the correlations. We note $\mathbf{\Lambda}$ the normalization matrix, the $\mathbf{C}_\alpha$ matrix is then defined :

$$\mathbf{C}_\alpha = \mathbf{\Lambda}_{\boldsymbol{\alpha}}\mathbf{L}_\alpha\mathbf{\Lambda}_\alpha$$

We make experiments minimizing both cost functions $\mathcal{J}$ and $\mathcal{J}^p$. Unfortunately, as they are not defined using the same $\mathbf{B}$ matrix, we cannot compare the results. First we will use the cost function $\mathcal{J}$ (6.26) in the following section 6.5.4. In 6.5.5, we will present the results for the minimization of $\mathcal{J}^p$ (6.31)

6.5.4 Numerical results without preconditionning

For the first experiments, we use the same parameters for high and low resolution data assimilation : $(\sigma_o = 0.6, \beta = \frac{1}{2\sqrt{2}}, \beta_1 = 10^3, \beta_2 = 10^{15}, \sigma_{b,\mathbf{h}} = 20, \sigma_{b,\mathbf{u}} = \sigma_{b,\mathbf{v}} = 0.1)$. The background on coarse grid is taken equal to the restriction of the fine grid one.

Monogrid minimizations

We search for the initial state $\mathbf{w}_0^a$ that minimizes the cost function (6.26). We then express the optimal initial solution $\mathbf{x}_0^a$ using the expression (6.27).
To minimize, we implement the non linear 4D-VAR and the incremental method, on fine grid, as references. For each one, we use Le Marechal's M1qn3 [21] as the optimization code.
We consider the decrease of the cost function according to computational time. In Figure 6.6 we observe the results for the two references algorithm, using the below parameters.

For the incremental method, inside each inner-loops, we made 3 iterations of M1QN3. Each symbol denotes a non linear misfit update. Between each symbol, we minimize a quadratic cost function.

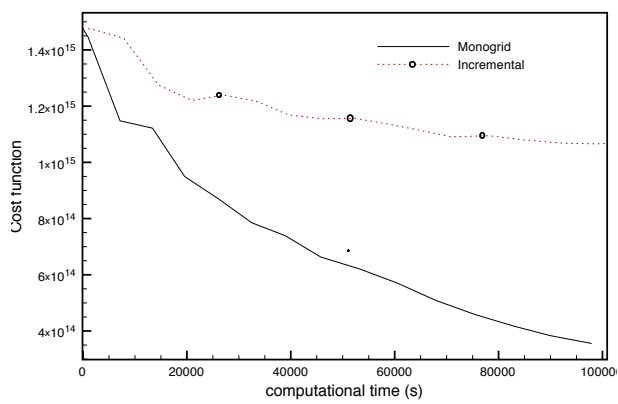

FIG. 6.6 – The decrease of the cost function for the non linear minimization and for the incremental method. Each symbol denotes a non linear misfit update.

The incremental method takes much more time than the monogrid to converge. Our model may not give a non-quadractic cost function such difficult to minimize so that the non-linear minimization is more efficient than the globlal linearizations of the incremental method. Indeed, it is worth noting that in realistic simulations, incremental approach is faster than non-quadratic minimization.

In Figure 6.7, we observe the RMS Error according to time integration

$$RMS(\mathbf{x}, \mathbf{x}^t) = \sqrt{\frac{1}{N_x} \sum_{j=0}^{N_x-1} (\mathbf{x}^j - (\mathbf{x}^j)^t)^2(t)} \tag{6.36}$$

for 2 values of initial state : the background and the analyzed state found after 15 iterations of non linear minimizations.

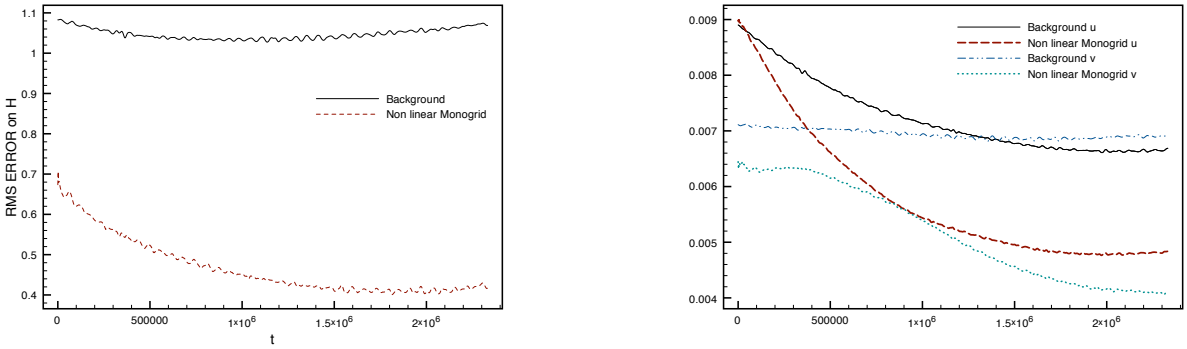

FIG. 6.7 – The RMS Error according to time integration, for **h** to the right and **u**, **v** to the left. Before (background) and after non linear fine grid minimization.

We can check that non linear minimization was truly effective to reduce the distance between the water height and the observed data, from the beginning of time integration to the end. It was also effective in decreasing the error made on non-observed data as **u** and **v**.

Non linear multigrid minimizations

Here we test the non linear multigrid algorithm, that is to say the FAS.

We use a line-search (with Goldstein conditions) for the coarse grid correction and we make a first coarse grid minimization of 10 iterations. Then every 2 fine grid iterations, we make a coarse grid correction with 10 coarse grid iterations. We call this algorithm "FMG FAS".

We compare it to the Non linear monogrid algorithm presented above and to "Coarse+Fine" that makes 10 coarse grid iterations first and then keeps on minimizing on the fine grid.
Results are showed Figure 6.8.

We note that the first coarse grid minimization has a real effect on the cost function decrease. But even after, the multigrid algorithm is faster than monogrid one. These results mean that, making first coarse grid minimization is not enough to solve large scales. As we are in non linear cases, we must take into account the large scales reintroduced by fine grid iterations. Multigrids is then a way to accelerate the solution of these remaining large frequencies. That is why coarse grid correction still has an effect after one first coarse grid minimization.

Incremental methods

On the same configuration, we compare the multigrid algorithms applied to incremental method. We test 3 algorithms, the monogrid incremental, the GN-MG and the multi-incremental. The results for incremental monogrid have been already introduced in Figure 6.6.
Each inner-loop of the GN-MG algorithm is made of one cycle of 5 coarse grid iterations followed by 2 fine grid iterations and then 5 coarse grid iterations.
The multi-incremental approach is made of outer-loops inside of each 5 coarse grid iterations

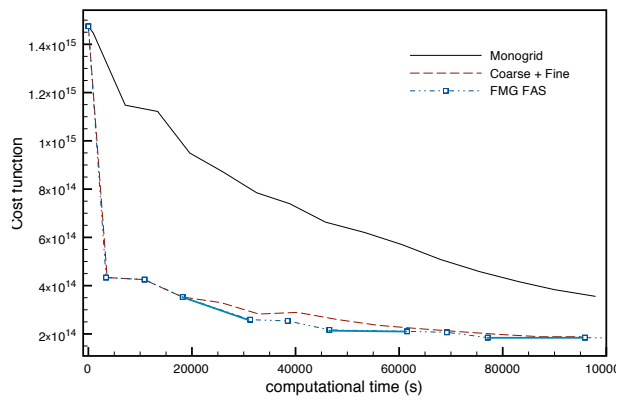

FIG. 6.8 – The decrease of the cost function for the non linear minimizations, Monogrid and multigrid algorithms. Every symbol of the FMG FAS denotes a fine grid calculation. The coarse grid steps are showed using segments.

were computed.
Results are shown Figure 6.9.

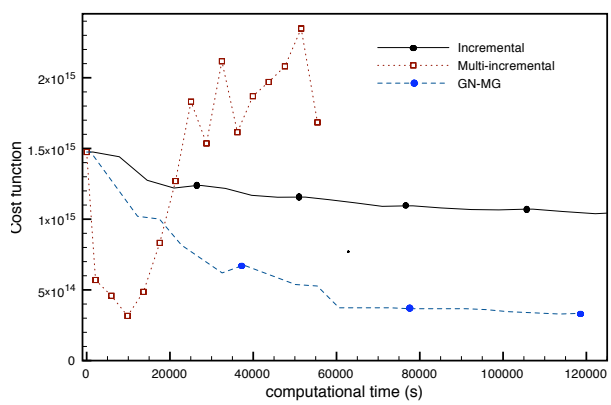

FIG. 6.9 – The decrease of the cost function for the incremental minimizations : Monogrid and multigrid algorithms. Each symbol denotes a non-linear misfit update.

The first remark is that the multi-incremental does not converge ; a common problem to their use (see Tremollet [61]). The first outer-loops enable to reduce the cost function, indeed it's about solving first on coarse grid to have a first good approximation of the fine grid solution. But doing too much outer-loops tends to increase the cost function. As the coarse grid system is not the fine grid one. Minimizing the coarse grid function does not insure us the fine grid one will also be reduced.
The GN-MG is much better. It takes more time to compute one outer-loop than the incremental monogrid. But in this example, it is clear that the three fine grid iterations used inside each outer loop for the incremental method are not enough to reach the same value of cost function as with the GN-MG.

6.5.5 Preconditionned cost function

In this section, we minimize the preconditionned cost function (6.31), using the same parameters as for the previous experiments. The spatial correlation length is set equal to $l = 50km$.

Galerkin relation

We introduce the use of the Galerkin relation for the $\mathbf{B}$ matrix. Indeed, due to the discrete properties of the diffusion operator $\mathbf{B}^c$ and $\mathbf{B}^f$ can be strongly different (and are in practice). Without using the Galerkin relation (6.24), the coarse grid correction step leads to a correction with inappropriate scaling.
For our preconditionned cost function, we write

$$\mathbf{x}_0^c = \mathbf{x}^{b,c} + (\mathbf{B}^c)^{\frac{1}{2}}\mathbf{w}_0^{p,c}$$

where

$$(\mathbf{B}^c)^{\frac{1}{2}}\mathbf{w}_0^{p,c} = I_f^c(\mathbf{B}^f)^{\frac{1}{2}}I_c^f\mathbf{w}_0^{p,c} \tag{6.37}$$

Note that this implies that if $(\mathbf{w}_0^p)^c$ is interpolated on the fine grid, the resulting fine grid initial state satisfies :

$$I_f^c\mathbf{x}_0^f = \mathbf{x}_0^c$$

Indeed $\mathbf{x}_0^f$ is then equal to $\mathbf{x}^{b,f} + (\mathbf{B}^f)^{\frac{1}{2}}I_c^f\mathbf{w}_0^{p,c}$ so that $I_f^c\mathbf{x}_0^f = \mathbf{x}_0^c$ since the two background are linked with $\mathbf{x}^{b,c} = I_f^c\mathbf{x}^{b,f}$.

As the operator (6.37) is applied to the initial state, calculating it on the fine grid instead of the coarse grid will not increase the cost of solution. So, we implement new versions of algorithms that transfers the coarse grid initial state to fine grid and calls the routines for the balance and the spatial correlations on the fine grid. Then we transfer back the results to the coarse grid.

Results

We test here 5 algorithms.
First we look at the non-linear minimizations. In Figure 6.10, the cost function decrease is shown for the Monogrid and the FAS algorithms.

The FAS algorithm uses cycles of 2 fine grid iterations followed by 10 coarse grid ones.

Even if we cannot compare with the previous results (because of the different $\mathbf{B}$ matrix), it is sure that the preconditionning is effective. Indeed, the monogrid algorithm is more efficient. A strong cost function decrease is done during the first iterations.
For the same reasons, and to the contrary to the unpreconditionned version, making coarse grid minimization before fine grid ones is not useful. So, the FAS algorithm begins with fine grid iterations.
Nevertheless, as seen on Figure 6.10 (right), the coarse grid corrections improve the cost function decrease and the multigrid FAS algorithm is faster than the monogrid one.

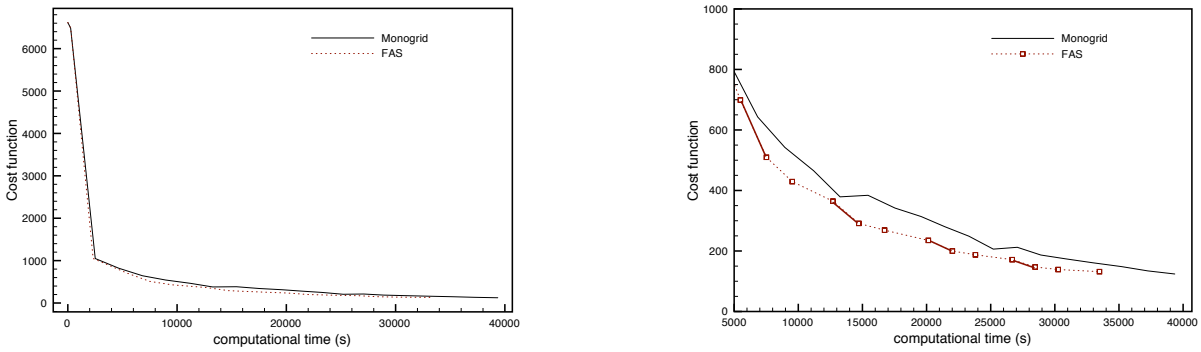

FIG. 6.10 – The decrease of the cost function, according to computational time for the Preconditionned Non-linear monogrid and FAS. At the right, zoom. For the FAS, each symbol denotes a fine grid calculation of cost function. The coarse grid steps are showed using segments.

A few remarks are worth to be made. For each minimizations, we use M1QN3 procedure. However, it is very sensitive to the choice of the different parameters. In practice, we have to do a minimum number of 2 or 3 iterations, while it may not be necessary. This optimization procedure could certainly be improved. Anyway a more suitable optimization method is desired.
We can think of using the coarse grid information to improve the knowledge of the fine grid Hessian. For example, the adaptation of the multigrid conjugate gradient algorithm presented in Pflaum [49] to non linear problems is a possibility.

We then test the incremental methods. The monogrid algorithm is made of 3 fine grid iterations inside each outer-loops.
GN-MG is componed with outer-loop of 2 cycles made with 2 fine grid iterations followed by 10 coarse grid ones.
Finally the Multi-incremental method is implemented with 10 coarse grid iterations for each outer-loops.
The decrease of the cost function is showed on Figure 6.11.

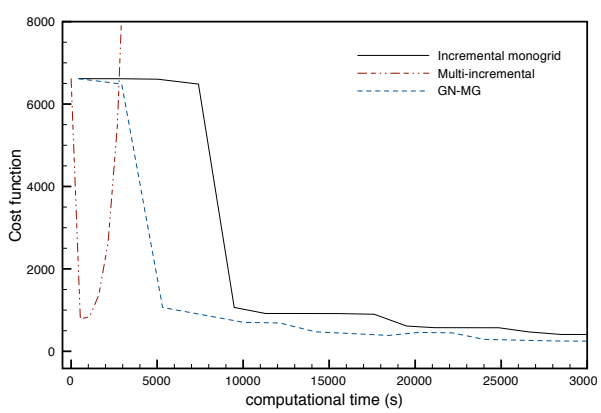

FIG. 6.11 – The decrease of the cost function for the Preconditionned incremental algorithms.

As for the results without preconditionning, the Multi-incremental does not converge.
The GN-MG and the monogrid incremental leads to the same values of cost function, GN-MG being faster than monogrid. The FAS is still better than the GN-MG, that was also the case of the incremental monogrid compared to the non linear monogrid method.

6.6 Conclusion

We introduced multigrid algorithms for solving a 4D variational data assimilation problem. We have implemented non linear multigrid methods, the Full Approximation Scheme (FAS) and the multigrid methods in the incremental version of the 4DVAR (GN-MG).
They have been tested on a Shallow water model, with two cost functions. The first one was defined without preconditionning whereas the second one was preconditionned.
The multigrid configurations, for all experiments, have shown that they can converge faster than monogrid algorithms. In our test set up, the non-linear minimizations were better than the incremental methods. This applies both to monogrid and multigrid configurations.

In the preconditionned case, we noticed the necessity of improving the multigrids behavior using the Galerkin relation for the $\mathbf{B}$ matrix, in order to make the coarse grid operators close to the fine grid ones.

In this paper, multigrid algorithms were showed to be more efficient than monogrid ones. However, we think that a lot of additional improvement can be achieved using specifically designed optimization procedure. This was not the case of the procedure used here (M1QN3). It should be customized or a multigrid optimization procedure as in [49] should be adapted.

CONCLUSION

Ignorer le passé,
c'est aussi raccourcir l'avenir.
- Julien Green -

Conclusion

L'assimilation de données en géophysique est un champ d'étude vaste dont nous n'avons exploré qu'une approche, l'assimilation variationnelle. De nombreuses recherches portent sur la validité de cette approche, en tentant de vérifier la capacité du modèle à décrire les observations choisies, et dénotent l'importance de la définition des différents opérateurs, en particulier des matrices de covariances d'erreurs. D'autres recherches abordent le problème d'un point de vue purement optimisation. C'est l'approche que nous avons choisie ici, en s'attachant à améliorer la résolution du système optimal traduisant la minimisation de l'écart du modèle aux observations. De par les méthodes multigrilles, nous avons cherché à diminuer le coût de ce calcul, principales limites à l'utilisation de l'assimilation de données à but prévisionnel, à haute résolution.

Nous sommes partis d'un modèle simple d'advection linéaire pour aider à la compréhension du fonctionnement. Peu à peu, des éléments se sont précisés. Le terme de régularisation de la fonction coût apporte de l'ellipticité au système, ce qui favorise l'utilisation des méthodes multigrilles. Cependant, l'ellipticité, même si préférable, n'est pas nécessaire. Les méthodes multigrilles offrent de grandes possibilités, puisque qu'elles peuvent être utilisées avec une certaine liberté dans la définition des opérateurs et des paramètres. Elles sont également applicables à une grande gamme de fonctions coût. Aussi, elles peuvent être vues comme un complément au préconditionnement traditionnel par la matrice de covariance des erreurs d'ébauche $\mathbf{B}$ ou comme une alternative lorsque celui-ci n'est pas efficace.

Forts de cette première expérience, nous avons alors étendu l'application à un modèle non linéaire, en l'occurence une équation de Burgers à une dimension, en développant deux algorithmes

multigrilles : le FAS, minimisant la fonction coût non quadratique ; et le GN-MG, minimisant chaque fonction coût quadratique d'un algorithme incrémental. Le premier résout directement le modèle non linéaire alors que le deuxième s'abstrait des difficultés dues aux non-linéarités, mais repose sur l'hypothèse de quasi-linéarité (hypothèse linéaire tangent). L'étude des différentes erreurs a montré que, pour notre exemple, la convergence des méthodes multigrilles est plus sensible à la linéarisation, qu'à l'approximation par la grille grossière. Le FAS se révèle donc plus robuste et avec moins de paramètres à fixer. Néanmoins il nécessite une bonne méthode d'optimisation adaptée aux fonctions coût non quadratiques ; alors que le GN-MG s'affranchit de cette obligation. Pour chaque algorithme, en fixant de manière adéquate les paramètres, le temps de calcul a pu être réduit d'un facteur 10 par rapport à une méthode monogrille.
Lorsqu'il n'entraîne pas de divergence de la méthode d'optimisation, le préconditionnement traditionnel est très efficace. Cependant appliquer une méthode multigrille FAS peut permettre d'encore accélérer la convergence.

Nous avons alors étendu l'application sur un modèle Shallow Water 2D pour lequel nous avons testé deux fonctions coût, avec et sans préconditionnement.
Dans le cas du FAS, la convergence des passages par la grille grossière n'est pas assurée, à cause des non-linéarités. Nous avons contourné cet obstacle en ajoutant une recherche linéaire à la correction par la grille grossière, ce qui nous a permis d'obtenir des résultats satisfaisants.
Dans ce cas plus complexe, le préconditionnement induit l'utilisation d'une matrice $\mathbf{B}$ dont la formulation est intrinsèquement lié à la grille de résolution sur laquelle elle est calculée. Ainsi, la matrice $\mathbf{B}$ de la basse résolution a été déduite de manière consistante en utilisant la formulation de Galerkin. Cela nous a permis de vérifier la propriété d'approximation essentielle pour la convergence des méthodes multigrilles.
Pour chaque cas, les méthodes multigrilles, bien paramétrées, ont convergé vers la solution haute résolution, plus rapidement que les méthodes monogrilles. La méthode multi-incrémentale, couramment utilisée en assimilation variationnelle opérationnelle, méthode qui calcule uniquement des corrections à basse résolution, est beaucoup moins robuste.

Perspectives

Ces premiers résultats ont prouvé l'efficacité des méthodes multigrilles à réduire le coût de calcul de l'assimilation de données variationnelles, pour des configurations simples. Nous pensons que leur intérêt peut être encore plus grand sur des cas plus complexes.

En effet, les méthodes multigrilles peuvent être utilisées sur des problèmes plus difficiles à préconditionner ; c'est-à-dire lorsque le préconditionnement est compliqué ou coûteux à mettre en place, ou lorsqu'il est inaccessible.
De tels problèmes se rencontrent lors de l'utilisation de formulations non diagonales de $\mathbf{R}$, la matrice des erreurs d'observations qui inclue les erreurs de représentativité. Complexifier la matrice $\mathbf{R}$ permet de mieux prendre en compte la dynamique des observations et de diminuer l'importance du terme de régularisation. L'assimilation devient alors moins dépendante de la connaissance de

l'ébauche et le préconditionnement par **B** perd de son intérêt. Or de tels matrices **R** vont bientôt être accessibles et utiles notamment dans le cadre de l'assimilation d'images satellites.
L'assimilation variationnelle de données avec contraintes faibles induit également un moindre intérêt du préconditionnement par **B**. En effet, le Hessien comporte de nombreux termes supplémentaires correspondants aux erreurs du modèle que l'on veut contrôler. Le conditionnement du Hessien dépend alors autant de ces termes que du terme à l'ébauche. Par contre, une configuration multigrille permettrait un contrôle local de l'erreur modèle à haute résolution, et un contrôle global à basse résolution ce qui rend le coût de calcul et la dimension du vecteur de contrôle sur la grille fine raisonnables.
Dans ces deux cas, nous espérons que les méthodes multigrilles sont une alternative efficace au préconditionnement pour aider à la réduction du coût de calcul.

Jusque là, nous avons abordé le problème des méthodes multigrilles essentiellement dans le but de diminuer le coût de calcul. L'objectif est en fait de pouvoir utiliser les méthodes d'assimilation de données variationnelles dans toutes leurs puissances, c'est à dire à haute résolution, et en prenant en compte plus de non linéarités. Les phénomènes fortement non linéaires complexifient la minimisation en augmentant le nombre de minima locaux. Pour contrer ce problème, l'assimilation est faite, jusqu'à présent, sur des modèles simplifiés, voire linéarisés. Nous pensons que l'utilisation de la méthode Full-Multigrid (FMG-FAS), en commençant la minimisation à basse résolution, s'affranchit de la présence de trop nombreux minima locaux ; tout en résolvant l'assimilation non linéaire à haute résolution.

L'efficacité des méthodes multigrilles peut aussi être augmentée en améliorant la méthode d'optimisation proprement dite.
Il suffit de construire une méthode M1QN3, spécialement adaptée aux méthodes multigrilles. En effet, cette méthode est basée sur le stockage, tout au long de la minimisation, des différentes approximations du produit Hessien-vecteur. Or, chaque passage par la grille grossière permet, dans le cas linéaire, d'approcher le Hessien de la grille fine par celui de la grille grossière. Il doit être possible d'utiliser cette information pour enrichir les approximations du Hessien utilisées par M1QN3.
De manière plus générale, nous pouvons aussi implémenter un algorithme d'optimisation, en adaptant aux cas non linéaires les algorithmes d'optimisation multi-résolutions de Pflaum [49] ou en exploitant les travaux de Gratton [25]. Il est possible aussi d'utiliser des algorithmes à région de confiance [24] pour s'assurer de la validité des hypothèses, de linéaire tangent pour l'incrémental et de consistance entre les systèmes haute et basse résolution pour le FAS.

Enfin, les méthodes multigrilles pourraient également être utilisées pour évaluer à un coût moindre la propagation d'erreurs, en utilisant l'information sur le Hessien donnée par chaque correction grille grossière. Cela permettrait d'implémenter des méthodes d'assimilation de données fournissant des renseignements sur la qualité des prévisions via l'information sur les erreurs.

BIBLIOGRAPHIE

[1] Amodei, L. Solution approchée pour un problème d'assimilation de données météorologiques avec prise en compte de l'erreur de modèle. *C.R. Acad. Sci. Paris 321*, 12 (1995), 1087–1094.

[2] Bennett, A. F. *Inverse methods in Physical Oceanography.* Cambridge University Press, 1992.

[3] Bennett, A. F. *Inverse Modeling of the Ocean and Atmosphere.* Cambridge University Press, 2002.

[4] Bennett, A. F., Leslie, L. M., Hagelberg, C. F., and Powers, P. E. Tropical cyclone prediction using a barotropic model initialized by a generalized inverse method. *Monthly Weather Review 121* (1993), 1714–1729.

[5] Brandt, A. Multi-level adaptive solutions to boundary-value problems. *Mathematics of Computation 31*, 138 (1977), 333–390.

[6] Brandt, A. Multi-level adaptive computations in fluid dynamics. *AIAA J. 18* (1980), 1165–1172.

[7] Brandt, A. Multigrid solutions to steady-state compressible navier-stokes equations i. In *Computing Methods in Applied Sciences and Engineering V* (1982), R. Glowinski and J. L. L. eds, Eds., Proc. Fifth International Symposium, Versailles, December 1981, pp. 407–423.

[8] Brandt, A. *Multigrid techniques 1984 Guide with applications to fluid dynamics.* Gesellschaft fuer Mathematik und Datenverarbeitung m.b.H. Bonn (GMD), St. Augustin (Germany), 1984.

[9] Briggs, W. L. *A multigrid tutorial.* Society for Industrial and Applied Mathematics, 1987.

[10] Courtier, P. Dual formulation of four-dimensional variational assimilation. *Q. J. R. Meteorol. Soc. 123* (1997), 2449–2461.

[11] Courtier, P., Thépaut, J. N., and Hollingsworth, A. A strategy for operational implementation of 4d-var, using an incremental approach. *Q. J. R. Meteorol. Soc. 123* (1994), 1367–1387.

[12] Da Silva, A., Pfaendtner, J., Guo, J., Sienkiewicz, M., and Cohn, S. Assessing the effects of data selection with dao's physicalspace statistical analysis system. In *Proceedings of the second international WMO symposium on assimilation of observations in meteorology and oceanography,Tokyo 13-17 March* (1995), vol. WMO.TD 651, pp. 273–278.

[13] De Leon, D. *Wavelets Operators Applied to Multigrid Methods.* PhD thesis, University of California, Los Angeles, 2000.

[14] Debreu, L., Simon, E., and Blayo, E. 4d variational data assimilation for locally nested models : optimality system and preliminary numerical experiments. *Int. J. Numer. Meth. Fluids 00 :1-6* (2000).

[15] Evensen, G. Using the extended kalman filter with a multilayer quasi-geostrophic model. *J. Geophys. Res. 97*, C11 (1992), 17905–17924.

[16] Evensen, G. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *J. Geophys. Res. 99*, C5 (1994), 10143–10162.

[17] Evensen, G. *Data assimilation. The Ensemble Kalman filter.* Springer, 2007.

[18] Ferreira, D., Marshall, J., and Heimbach, P. Estimating eddy stresses by fitting dynamics to observations using a residual-mean ocean circulation model and its adjoint. *Journal of Physical Oceanography 35* (2005), 1891–1910.

[19] Gandin, L. S. *Objective analysis of meteorological fields.* Gidrometeorol. Izd., Leningrad,, 1963.

[20] Ghil, M., Cohn, S., Tavantzis, J., Bube, K., and Isaacson, E. Applications of estimation theory to numerical weather prediction. In *Dynamic Meteorology : Data assimilation Methods*, L. Bengtsson, M. Ghil, and E. Kallénn, Eds. Springer Verlag, 1981, pp. 139–224.

[21] Gilbert, J.-C., and Lemaréchal, C. Some numerical experiments with variable storage quasi-newton algorithms. *Mathematical Programming B25* (1989), 407–435.

[22] Goedecker, S., and Chauvin, C. Combining multigrid and wavelet ideas to construct more efficient multiscale algorithms for the solution of poisson's equation. *J. of Theoritical and Computational Chemistry 2*, 483 (2003).

[23] Gratton, S., Lawless, A. S., and Nichols, N. K. Approximate gauss-newton methods for non-linear least square problems. *SIAM J. Optimization 18* (2007), 106–132.

[24] Gratton, S., Sartenaer, A., and Toint, P. Recursive trust-region methods for multiscale nonlinear optimization. *SIAM J. on Optimization 19*, 1 (2008), 414–444.

[25] Gratton, S., and Toint, P. Approximate invariant subspaces and quasi-newton optimization. *Optimization Methods and Software 25*, 4 (2010), 507–529.

[26] Hackbusch, W. *Multi-grid Methods and Applications.* Springer Series in Computational Mathematics, 1985.

[27] Hascoët, L., and Pascual, V. Tapenade 2.1 user's guide. Technical Report 0300, INRIA, http ://www.inria.fr/rrrt/rt-0300.html, 2004.

[28] Hemker, P. A note on defect correction processes with an approximate inverse of deficient rank. *Appl. Math. Comp. 8* (1982), 137–139.

[29] Hemker, P. On the order of prolongations and restrictions in multigrid procedures. *Journal of Computational and Applied Mathematics 32* (1990), 423–429.

[30] Hemker, P. Lecture notes on defect correction. Tech. rep., 2001.

[31] Holland, W. R., and Malanotte-Rizzoli, P. Along track assimilation of altimeter data into an ocean circulation model : space versus time resolution studies. *J. Phys. Oceanogr. 19*, 1507-1534 (1989).

[32] Ide, K., Courtier, P., Ghil, M., and Lorenc, A. C. Unified notation for data assimilation : Operational, and variational. *J. Met. Soc. of Japan 75*, 1B (1997), 181–189.

[33] Ifremer. www.ifremer.fr. *Institut français de recherche pour l'exploitation de la mer*.

[34] Jones, R. H. Optimal estimation of initial conditions for numerical analysis. *J. Atmos. Sci. 22* (1965), 658–1965.

[35] Kalman, R. E. A new approach to linear filtering and prediction problems. *Transactions of the ASME-Journal of Basic Engineering 82*, Series D (1960), 33–45.

[36] Kettler, R. Analysis and comparison of relaxation schemes in robust multigrid and preconditioned conjugate gradient methods. In *Multigrid methods, Proc. conference held at Köln-Porz* (1981), W. Hackbush and U. T. eds., Eds.

[37] Krüger, J. Simulated annealing : a tool for data assimilation into an almost steady state model state. *American Meteorological Society 23* (1993), 679–688.

[38] Le Dimet, F.-X. A general formalism of variational analysis. Tech. rep., CIMMS, 1982.

[39] Le Dimet, F.-X., and Talagrand, O. Variational algortihms for analysis and assimilation of meteorological observations : Therotical aspects. *Tellus 38*, A (1986), 97–110.

[40] Lewis, R. M., and Nash, S. G. Model problems for the multigrid optimization of systems governed by differential equations. *SIAM J. SCI. COMPUT. 26*, 6 (2005), 1811–1837.

[41] Lions, J. L. *Contrôle optimal des systèmes gouvernès par des équations aux dérivées partielles.* Dunod, Paris, 1968.

[42] Lorenz, E. N. Deterministic non periodic flow. *Atmos. Sci. 20* (1963), 130–141.

[43] McCormick, S. F. An algebraic interpretation of multigrid methods. *SIAM J. Num. Anal. 19* (1982), 548–560.

[44] Miller, A. J., and Cornuelle, B. D. Forecasts from fits of frontal fluctuations. *Dynamics of atmosphere and oceans 29* (1999), 305–333.

[45] Miller, R. N. Toward the application of the kalman filter to regional open ocean modeling. *J. Phys. Oceanogr. 16* (1986), 72–86.

[46] Miller, R. N., Carter, Everett F., J., and Blue, S. T. Data assimilation into nonlinear stochastic models. *Tellus 51 A* (1999), 167–194.

[47] Moore, A. M. Data assimilation in a quasi-geostrophic open-ocean model of the gulf stream region using the adjoint method data assimilation in a quasi-geostrophic open-ocean model of the gulf stream region using the adjoint method. *J. Phys. Oceanogr. 21*, 382-427 (1991).

[48] Nash, S. G. A multigrid approach to discretized optimization problems. *Journal of Optimization Methods and Software 14* (2000), 99–116.

[49] Pflaum, C. A multigrid conjugate gradient method. *Applied Numerical Mathematics 58* (2008), 1803–1817.

[50] Pham, D. T., Verron, J., and Roubaud, M.-C. A singular evolutive extended kalman filter for data assimilation in oceanography. *Journal of Marine Systems 16*, 3-4 (1998), 323–340.

[51] Press, W. H., Flannery, B. P., and Teukolsky, S. A. *Numerical Recipes*. Cambridge University Press, 1992.

[52] Robert, R., and Rosier, C. Long range predictability of atmospheric flows. *Nonlinear processes in geophysics 8* (2001), 55–67.

[53] Sasaki, Y. A fundamental study of the numerical prediction based on the variational principle. *J. Met. Soc. of Japan 33* (1955), 262–275.

[54] Shchepetkin, A. F., and McWilliams, J. C. The regional oceanic modeling system (roms) : a split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling 9* (2005), 347–404.

[55] Simon, E. *Assimilation variationnelle de données pour des modèles emboîtés*. PhD thesis, Université Joseph Fourier, Grenoble I, 2007.

[56] Southwell, R. V. Stress-calculation in frameworks by the method of "systematic relaxation of constraints. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 151*, 872 (1935), 56–95.

[57] Southwell, R. V. Relaxation methods in theoretical physics. Tech. rep., Oxford University Press, 1946.

[58] Ta'asan, S. One shot methods for optimal control of distributed parameters systems i : Finite dimensional control. Tech. Rep. No 91-2, ICASE Report, 1991.

[59] Ta'asan, S. *Multigrid one-shot methods and Design Strategy*. Carnegie Mellon University, 1997.

[60] Thépaut, J. N., and Courtier, P. Four-dimensional variational data assimilation using the adjoint of a multilevel primitive-equation model. *Q. J. R. Meteorol. Soc. 117* (1991), 1225–1254.

[61] Trémolet, Y. Incremental 4d-var convergence study. *Tellus 59A* (2007), 706–718.

[62] Trottenberg, U., Oosterlee, C. W., and Schüller, A. *Multigrid*. Academic Press, 2001.

[63] Veersé, F., and Thépaut, J. N. Multiple-truncation incremental approach of four-dimensional variational data assimilation. *Q. J. R. Meteorol. Soc. 124* (1998), 1889–1908.

[64] Verlaan, M., and Heemink, A. W. Tidal flow forecasting using reduced rank square root filters. *Stoch. Hydrol. Hydraul. 11*, 5 (1997), 349–368.

[65] Verron, J. Nudging satellite altimeter data into quasi-geostrophic ocean model. *J. Geophys. Res. 97* (1992), 7479–7491.

[66] Vidard, P. A. *Vers une prise en compte des erreurs modèle en assimilation de données 4D-variationnelle. Application à un modèle réaliste d'océan.* PhD thesis, Université Joseph Fourier, Grenoble I, 2001.

[67] Wang, Z., Navon, I. M., Le Dimet, F.-X., and Zhou, X. The second order adjoint analysis : theory and applications. *Meteorology and Atmospheric Physics 50* (1992), 3–20.

[68] Weaver, A. T., and Courtier, P. Correlation modelling on the sphere using a generalized diffusion equation. *Q. J. R. Meteorol. Soc. 127* (2001), 1815–1846.

[69] Weaver, A. T., and Ricci, S. Constructing a background-error correlation model using generalized diffusion operators. In *ECWMF procreedings on "Recent developments in data assimilation for atmosphere and ocean"* (2003).

[70] Yavneh, I., and Dardyk, G. A multilevel nonlinear method. *SIAM J. SCI. COMPUT. 28*, 1 (2006), 24–46.

Résumé : Depuis ces trente dernières années, les systèmes d'observation de la Terre et les modèles numériques se sont perfectionnés et complexifiés pour nous fournir toujours plus de données, réelles et numériques. Ces données, de nature très diverse, forment maintenant un ensemble conséquent d'informations précises mais hétérogènes sur les structures et la dynamique des fluides géophysiques. Dans les années 1980, des méthodes d'optimisation, capables de combiner les informations entre elles, ont permis d'estimer les paramètres des modèles numériques et d'obtenir une meilleure prévision des courants marins et atmosphériques. Ces méthodes puissantes, appelées assimilation variationnelle de données, peinent à tirer profit de la toujours plus grande complexité des informations de par le manque de puissance de calcul disponible. L'approche, que nous développons, s'intéresse à l'utilisation des méthodes multigrilles, jusque là réservées à la résolution de systèmes d'équations différentielles, pour résoudre l'assimilation haute résolution de données. Les méthodes multigrilles sont des méthodes de résolution itératives, améliorées par des corrections calculées sur des grilles de plus basses résolutions. Nous commençons par étudier dans le cas d'un modèle linéaire la robustesse de l'approche multigrille et en particulier l'effet de la correction par grille grossière. Nous dérivons ensuite les algorithmes multigrilles dans le cadre non linéaire. Les deux types d'algorithmes étudiés reposent d'une part sur la méthode de Gauss Newton multigrille et d'autre part sur une méthode sans linéarisation globale : le Full Approximation Scheme (FAS). Ceux-ci sont appliqués au problème de l'assimilation variationnelle de données dans le cadre d'une équation de Burgers 1D puis d'un modèle Shallow-water 2D. Leur comportement est analysé et comparé aux méthodes plus traditionnelles de type incrémentale ou multi-incrémentale.

Mots-clés : Assimilation variationnelle de données, méthodes multigrilles, contrôle optimal, modèle multirésolution, modélisation numérique océanographique, méthodes numériques.

Abstract : For these last thirty years, earth observation and numerical models improved greatly and provide now a huge amount of accurate, yet heterogeneous, information on geophysics fluids dynamics and structures. Optimization methods from the eighties called "variational data assimilation" are capable of merging information from different sources. They have been used to estimate the parameters of numerical models and better forecast oceanic and atmospheric flows. Unfortunately, these powerful methods have trouble making benefit of always more complex information, suffering from the lack of available powerful calculators.
The approach developed here, focuses on the use of multigrid methods, that are commonly used in the context of differential equations systems, to solve high resolution data assimilation. Multigrid methods are iterative methods improved by the use of feedback corrections evaluated on coarse resolution. First in the case of linear assimilation, we study the robustness of multigrid approach and the efficiency of the coarse grid correction step. We then apply the multigrid algorithms on a non linear 1-D Burgers equation and on a 2-D Shallow-Water model. We study two types of algorithms, the Gauss Newton Multigrid, which lays on global linearization, and the Full Approximation Scheme. Their behavior is compared to more traditional approaches as incremental and multi-incremental ones.

Keywords : Variational data assimilation, multigrid methods, optimal control, multiresolution models, numerical ocean modelling, numerical methods.