



HAL
open science

Étude et développement d'algorithmes d'assimilation de données variationnelle adaptés aux modèles couplés océan-atmosphère

Rémi Pellerej

► **To cite this version:**

Rémi Pellerej. Étude et développement d'algorithmes d'assimilation de données variationnelle adaptés aux modèles couplés océan-atmosphère. Optimisation et contrôle [math.OC]. Université Grenoble - Alpes, 2018. Français. NNT: . tel-01806281v1

HAL Id: tel-01806281

<https://theses.hal.science/tel-01806281v1>

Submitted on 1 Jun 2018 (v1), last revised 17 Oct 2018 (v2)

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.

THÈSE

Pour obtenir le grade de

**DOCTEUR DE LA COMMUNAUTE UNIVERSITE
GRENOBLE ALPES**

Spécialité : **Mathématiques Appliquées**

Arrêté ministériel : 25 mai 2016

Présentée par

Rémi PELLEREJ

Thèse dirigée par **Arthur VIDARD**
et co-encadrée par **Florian LEMARIÉ**

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

Étude et développement d'algorithmes d'assimilation de données variationnelle adaptés aux modèles couplés océan- atmosphère

Thèse soutenue publiquement le **26 Mars 2018**,
devant le jury composé de :

Monsieur Éric BLAYO

Professeur, Université Grenoble Alpes, Président du jury

Monsieur Anthony WEAVER

Chercheur Senior, CERFACS, Rapporteur

Monsieur Didier AUROUX

Professeur, Université de Nice Sophia Antipolis, Rapporteur

Monsieur Ehouarn SIMON

Maître de conférence, ENSEEIHT, Examineur

Monsieur Arthur VIDARD

Chargé de recherche, INRIA, Directeur de thèse

Monsieur Florian LEMARIÉ

Chargé de recherche, INRIA, Co-encadrant de thèse



REMERCIEMENTS

Tout d'abord à Anthony Weaver et Didier Auroux d'avoir accepté de rapporter ces travaux de thèse, ainsi qu'à Ehouarn Simon et Eric Blayo pour avoir participé à mon jury. Merci pour vos remarques judicieuses qui m'ont permis de peaufiner ce manuscrit.

Un grand merci à Arthur et Florian pour avoir encadré cette thèse et m'avoir soutenu au cours de ces trois années (et un peu plus). J'imagine que ce n'était pas de tout repos pour vous non plus ! Je souhaite également remercier Yannick pour m'avoir donné toutes les clefs nécessaires afin de dompter OOPS. Cela ne fut pas une mince affaire.

Je remercie aussi toute l'équipe Airsea, Anne et Annie, Laurence et Suzan pour leur aide précieuse, sans qui ces trois années ne se passeraient pas aussi bien. Merci aussi à Adrien, JB, Lionel, Roland, Reda, Meriem et bien d'autres, avec qui nous avons passé de bons moments. Merci à Cathy pour toutes ces sorties sur les routes et ton soutien sans faille au LJK.

Je souhaite aussi remercier tout particulièrement Nelson, Alex et Charles, tmtc.

Merci enfin à tous mes proches, et surtout à toi, Margaux, pour m'avoir si bien épaulé et accompagné durant cette thèse, et de continuer à le faire... merci pour tout.

RÉSUMÉ

La qualité des prévisions météorologiques repose principalement sur la qualité du modèle utilisé et de son état initial. Cet état initial est reconstitué en combinant les informations provenant du modèle et des observations disponibles en utilisant des techniques d'assimilation de données. Historiquement, les prévisions et l'assimilation sont réalisées dans l'atmosphère et l'océan de manière découplée. Cependant, les centres opérationnels développent et utilisent de plus en plus des modèles couplés océan-atmosphère. Or, assimiler des données de manière découplée n'est pas satisfaisant pour des systèmes couplés. En effet, l'état initial ainsi obtenu présente des inconsistances de flux à l'interface entre les milieux, engendrant des erreurs de prévision. Il y a donc besoin d'adapter les méthodes d'assimilation aux systèmes couplés. Ces travaux de thèse s'inscrivent dans ce contexte et ont été effectués dans le cadre du projet FP7 ERA-Clim2, visant à produire une réanalyse globale du système terrestre.

Dans une première partie, nous introduisons les notions d'assimilation de données, de couplage et les différentes méthodologies existantes appliquées au problème de l'assimilation couplée. Ces méthodologies n'étant pas satisfaisantes en terme de qualité de couplage ou de coût de calcul, nous proposons, dans une seconde partie, des méthodes alternatives. Nous faisons le choix de méthodes d'assimilation basées sur la théorie du contrôle optimal. Ces alternatives se distinguent alors par le choix de la fonction coût à minimiser, des variables contrôlées et de l'algorithme de couplage utilisé. Une étude théorique de ces algorithmes a permis de déterminer un critère nécessaire et suffisant de convergence dans un cadre linéaire. Pour conclure cette seconde partie, les performances des différentes méthodes introduites sont évaluées en terme de qualité de l'analyse produite et de coût de calcul à l'aide d'un modèle couplé linéaire 1D. Dans une troisième et dernière partie, un modèle couplé non-linéaire 1D incluant des paramétrisations physique a été développé et implémenté dans OOPS (*Object-Oriented Prediction System*) qui est une surcouche logicielle permettant la mise en œuvre d'un ensemble d'algorithmes d'assimilation de données. Nous avons alors pu évaluer la robustesse de nos algorithmes dans un cadre plus réaliste, et conclure sur leurs performances vis à vis de méthodes existantes. Le fait d'avoir développé nos méthodes dans le cadre de OOPS devrait permettre à l'avenir de les appliquer aisément à des modèles réalistes de prévision. Nous exposons enfin quelques perspectives d'amélioration de ces algorithmes.

ABSTRACT

In the context of operational meteorology and oceanography, forecast skills heavily rely on the model used and its initial state. This initial state is produced by a proper combination of model dynamics and available observations via data assimilation techniques. Historically, numerical weather prediction is made separately for the ocean and the atmosphere in an uncoupled way. However, in recent years, fully coupled ocean-atmosphere models are increasingly used in operational centres. Yet the use of separated data assimilation schemes in each medium is not satisfactory for coupled problems. Indeed, the result of such assimilation process is generally inconsistent across the interface, thus leading to unacceptable artefacts. Hence, there is a strong need for adapting existing data assimilation techniques to the coupled framework. This PhD thesis is related to this context and is part of the FP7 ERA-Clim2 project, which aim to produce an earth system global reanalysis.

We first introduce data assimilation and model coupling concepts, followed by some existing algorithms of coupled data assimilation. Since these methods are not satisfactory in terms of coupling strength or numerical cost, we suggest, in a second part, some alternatives. These are based on optimal control theory and differ by the choice of the cost function to minimize, controlled variable and coupling algorithm used. A theoretical study of these algorithms exhibits a necessary and sufficient convergence criterion in a linear case. To conclude about this second part, the different methods are compared in terms of analysis quality and numerical cost using a 1D linear model. In a third part, a 1D non-linear model with subgrid parametrizations was developed and implemented in OOPS (Object-Oriented Prediction System), a software overlay allowing the implementation of a set of data assimilation algorithms. We then assess the robustness of the different algorithms in a more realistic case, and concluded about their performances against existing methods. By implementing our methods in OOPS, we hope it should be easier to use them with operational forecast models. Finally, we expose some prospects for improving these algorithms.

Table des matières

Introduction	1
I État des lieux	7
1 Méthodes usuelles d'assimilation de données	11
1.1 Principe de l'assimilation de données	11
1.2 Les méthodes stochastiques	17
1.3 Les méthodes variationnelles	21
2 Le couplage au sein des modèles de prévision	31
2.1 Modélisation et physique de l'océan et de l'atmosphère	32
2.2 Le problème du couplage océan-atmosphère	37
2.3 Méthodes de couplage usuelles et méthode de Schwarz	38
3 Histoire du développement de l'assimilation de données couplée	43
3.1 Motivations et vocabulaire de l'assimilation de données couplée	44
3.2 Méthodes d'assimilation de données introduites au fil du temps	46
3.3 Objectifs	54
Synthèse des notations	55
II Étude et développement d'algorithmes d'assimilation de données variationnelle couplée	59
4 Introduction de quelques méthodes d'assimilation couplée	63
4.1 Méthodes d'assimilation de données couplée	64
4.2 Étude de convergence des algorithmes d'assimilation couplée	74
5 Évaluation numérique de quelques algorithmes d'assimilation couplée dans un cadre linéaire	85
5.1 Description d'un modèle simple colonne de diffusion	85
5.2 Assimilation d'une observation colonne unique	88
5.3 Discussions et perspectives	99

III Évaluation d'algorithmes d'assimilation avec un modèle couplé colonne au sein d'un système de prévision complexe	101
6 Un modèle couplé simple colonne avec paramétrisations sous-maille	105
6.1 Description du modèle couplé colonne	106
6.2 Choix d'implémentation du modèle	112
6.3 Stratégies de couplage	116
7 Implémentation de modèles et d'algorithmes dans OOPS	119
7.1 Introduction à OOPS	120
7.2 Implémentation du modèle couplé simple colonne dans OOPS	121
7.3 Implémentation du terme J^s dans OOPS	127
7.4 Test du gradient et évaluation de la fonction coût linéaire	129
8 Évaluation numérique de quelques algorithmes d'assimilation couplée dans un cadre non-linéaire	131
8.1 Création d'un jeu de données	131
8.2 Description des expériences numériques	141
8.3 Résultats et discussions	146
Conclusions et perspectives	161
Annexes	
A Liste des classes d'interfaces de OOPS et de leurs méthodes à implémenter	169
Bibliographie	175

Introduction

Le système terrestre et son étude

La terre est un système complexe composé de nombreux milieux interagissant entre eux comme l'atmosphère et l'océan (voir figure I), mais aussi la glace de mer, les continents etc.

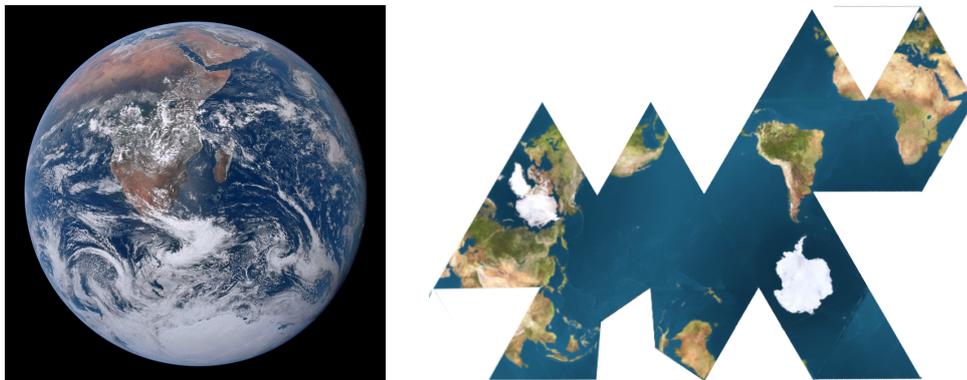


FIGURE I – À gauche : la terre vue de l'espace par un satellite permettant de visualiser la circulation atmosphérique grâce aux nuages. À droite : un planisphère dont la représentation met en valeur l'importance de l'océan à la surface de la terre

Historiquement, l'Homme a cherché à comprendre comment évoluaient ces différentes composantes du système terrestre. Il est ainsi parvenu à décrire l'état de ces milieux par l'intermédiaire de différentes lois et paramétrisations physiques et mathématiques que l'on appelle des modèles. Un modèle est une représentation d'un système permettant de le décrire via un ensemble d'équations. On pourrait dire qu'il s'agit d'un objet mathématique abstrait permettant de représenter le réel. Or, si l'on est capable de représenter l'évolution passée ou présente d'un système avec un modèle, on est en droit de penser que l'on peut estimer son état futur. C'est ainsi que les premiers modèles de prévisions ont été créés, tentant d'estimer l'évolution des différents paramètres (températures, vitesses...) de l'atmosphère, de l'océan, des continents etc. Aujourd'hui, ces modèles de prévision sont de plus en plus précis et sont utiles dans de nombreux secteurs tels que l'industrie, la défense ou encore l'agriculture.

Modélisation d'un système

L'évolution temporelle d'un système, tel que l'atmosphère, est décrite par un modèle généralement constitué d'un ensemble d'équations. Basiquement, nous pouvons représenter un modèle \mathcal{M} comme étant constitué d'une équation différentielle et d'une condition initiale :

$$\mathcal{M} : \begin{cases} \frac{\partial \mathbf{x}}{\partial t} = f(\mathbf{x}, t) \\ \mathbf{x}(t_0) = \mathbf{x}_0 \end{cases}$$

où $\mathbf{x}(t)$ est un vecteur représentant l'état du système au cours du temps, par exemple la température, la vitesse du vent ou encore l'humidité de l'air. \mathbf{x}_0 est la condition initiale à l'instant $t = t_0$ et f est une fonction de l'état et du temps. Notons \mathcal{S} l'espace fonctionnel de $\mathbf{x}(t)$. Nous considérons que $\mathbf{x}_0 \in \mathcal{S}$, $\mathcal{M} : \mathcal{S} \rightarrow \mathcal{S}$ et que $\mathbf{x}(t) = \mathcal{M}_{t_0 \rightarrow t}(\mathbf{x}_0)$ représente l'intégration du modèle jusqu'au temps t , et correspond à la prévision de l'état du système au temps t .

Réalisation d'une prévision

La prévision au temps t , notée $\mathbf{x}(t)$, est le résultat d'une suite de processus appelée *chaîne de prévision* (schématisée par la figure II), qui peut être résumée en deux étapes :

- la première consiste à initialiser correctement le modèle de prévision en lui procurant un état initial (précédemment \mathbf{x}_0) le plus juste possible ainsi que de bonnes conditions aux frontières. Cet état initial est l'*analyse* et doit se rapprocher le plus possible de l'*état vrai*, c'est à dire l'état réel du système (qui n'est pas connu). Cette analyse est créée à partir de la connaissance a priori du système (souvent la prévision précédente) que l'on appelle *ébauche* ou *background* ainsi que des observations disponibles. On appelle cette étape l'**assimilation de données**.
- la deuxième étape consiste ensuite à **intégrer le modèle** à partir de l'analyse pour créer une nouvelle prévision.

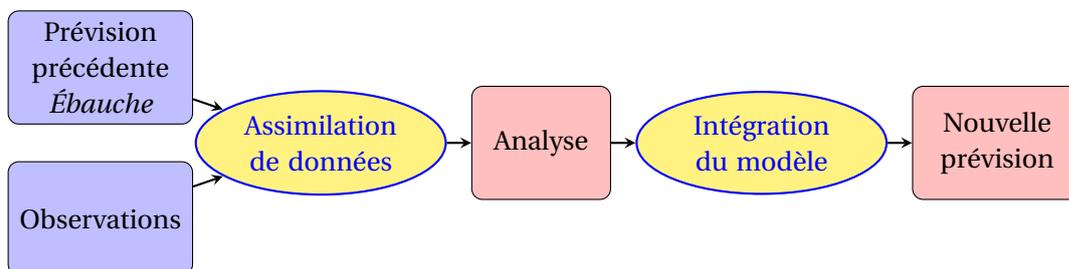


FIGURE II – Schématisation de la chaîne d'exécution lors d'une prévision météorologique

La qualité d'une prévision dépend de plusieurs facteurs, dont deux principaux :

- la qualité du **modèle** de prévision,
- la qualité de l'**assimilation de données** pour fournir une *bonne* analyse au modèle.

L'étude du système terrestre

Via les analyses

Les analyses ne servent pas seulement à fournir un état initial à un modèle pour réaliser une prévision. En effet, il s'agit également de la meilleure estimation de l'état d'un système que l'on dispose à un moment donné. Ainsi, en concaténant plusieurs analyses réalisées, il est possible d'obtenir une évolution temporelle de ce système. Ces analyses temporelles permettent par exemple d'étudier le comportement de l'atmosphère ou de l'océan à des échelles régionales, mais présentent quelques limites. En effet, au moment de la création d'une analyse, de nombreuses observations ne sont pas encore disponibles (relevés en cours, retards de transmission, etc). Or, l'analyse est tout de même produite avec les observations incomplètes puisque les prévisions doivent être fournies le plus rapidement possible. De plus, le modèle de prévision et le système d'assimilation évoluant en permanence, une analyse créée aujourd'hui sera généralement de meilleure qualité qu'une analyse créée il y a dix ans à partir d'un même jeu de données. Le manque d'observations et l'évolution du système d'assimilation impliquent que l'analyse n'est pas le meilleur candidat pour décrire l'évolution d'un système sur de longues périodes temporelles. C'est pourquoi les réanalyses, que nous allons décrire, seront généralement préférées aux analyses.

Les réanalyses préférées aux analyses

Une réanalyse est une analyse créée sur une période temporelle passée avec l'ensemble des observations disponibles et un unique système d'assimilation de données. Nous pouvons par exemple citer la réanalyse atmosphérique ERA-20C¹ du Centre Européen pour les Prévisions Météorologiques à Moyen Terme (CEPMMT) qui couvre la période entre Janvier 1900 et Décembre 2010. Utiliser le même système d'assimilation de données sur toute la période temporelle permet de comparer de manière objective l'état du système à deux moments différents. En effet, les différences observées par exemple dans ERA-20C entre le mois de janvier 2005 et janvier 2010 ne seront pas dues à un changement du système d'assimilation, comme cela pourrait être le cas entre deux analyses à ces deux dates. De plus, a contrario d'une analyse, une réanalyse pourra bénéficier de l'intégralité des observations réalisées sur la période temporelle considérée. Enfin, un dernier argument en faveur des réanalyses est la cohérence temporelle de ces dernières par rapport aux analyses. En effet, les analyses ne sont effectuées qu'à certains moments, il n'y a donc pas de continuité au cours du temps. Les réanalyses sont quant à elles effectuées sur des fenêtres temporelles complètes, avec un même système, permettant alors d'obtenir cette continuité. C'est pour l'ensemble de ces raisons que les réanalyses seront souvent préférées aux analyses pour étudier les systèmes atmosphériques ou océaniques.

Les enjeux de l'assimilation de donnée couplée

La qualité d'une analyse, d'une réanalyse ou d'une prévision est en partie conditionnée par la qualité du système d'assimilation et du modèle. Des efforts sont donc fournis par les différents centres de prévisions pour perfectionner leurs différents systèmes. En ce qui concerne les modèles de prévision d'atmosphère et d'océan, il a pu être mis en évidence que la prise en

1. Réanalyse effectuée par le Centre Européen de Prévisions Météorologiques à Moyen Terme. Voir <http://www.era-clim.eu/> pour plus d'informations.

compte des interactions à l'interface air-mer est un facteur déterminant pour la qualité des prévisions (voir par exemple Cane *et al.* (1986) ou Lemarié *et al.* (2014)). En effet, le couplage permet une rétroaction de l'atmosphère sur l'océan et vice-versa lorsqu'un changement survient dans un des deux milieux (hausse de température, précipitations, etc). Les systèmes peuvent alors s'équilibrer au cours de l'intégration du modèle, permettant de produire une prévision de meilleure qualité, notamment proche de l'interface air-mer. C'est pourquoi de nombreux modèles couplés océan-atmosphère sont développés ces dernières années et seront certainement bientôt utilisés par les centres météorologiques pour réaliser leurs prévisions à court terme.

Les systèmes d'assimilation vont devoir, en toute logique, être adaptés à ces nouveaux modèles couplés afin de leur fournir une analyse en accord avec leur dynamique couplée. En effet, si l'état initial n'est pas en équilibre à l'interface, le modèle couplé cherchera à rétablir cet équilibre, dégradant par conséquent la prévision (Mulholland *et al.*, 2015). Nous appellerons assimilation de données couplée le fait de produire une analyse en équilibre à l'interface air-mer. De plus, l'utilisation des méthodes d'assimilation couplée peut être utile dans le but d'augmenter le réalisme des réanalyses. En effet, ces méthodes devraient permettre de mieux représenter la dynamique des milieux à l'interface air-mer et leurs interactions. L'assimilation couplée offre également la possibilité qu'une observation dans un milieu ait une influence sur l'autre milieu. C'est pourquoi les premières réanalyses couplées sont actuellement créées et que le développement de systèmes d'assimilation de données couplée est devenu un des enjeux majeurs des centres météorologiques ces dernières années. Nous pouvons citer par exemple le projet européen FP7 ERA-Clim2², visant à reproduire une réanalyse couplée globale du système terrestre. La réanalyse couplée CERA-20C³ a ainsi été produite dans le cadre de ce projet européen. Ces travaux de thèse ont été financés dans le cadre du projet ERA-Clim2, au sein du *work package* n°2 portant sur le développement de nouvelles méthodes d'assimilation de données couplées. L'ensemble de ces enjeux ont pu être cités au cours de la dernière conférence internationale sur l'assimilation de données couplée⁴, à laquelle a été présentée une partie de ces travaux. Notons que de nombreux autres enjeux existent pour l'assimilation couplée, que nous pouvons retrouver dans le rapport publié par l'Organisation Météorologique Mondiale à la suite de cette conférence (Penny *et al.*, 2017).

Objectifs et organisation du manuscrit

De nombreuses méthodes d'assimilation de données couplée ont été développées ces dernières années, utilisant divers modèles couplés. Nous pouvons cependant regretter que dans la plupart des cas, peu d'attention est portée à la qualité du couplage au sein du modèle. Or, la majorité des modèles couplés utilisés ne prennent en compte que partiellement les interactions à l'interface air-mer. L'analyse créée n'est donc que partiellement couplée. Il existe des méthodes de couplage itératives permettant de coupler fortement les deux milieux. L'atmosphère et l'océan sont alors en équilibre parfait à l'interface air-mer. Ces méthodes de couplage sont cependant jugées trop coûteuses pour les centres de prévisions puisqu'elles nécessitent de réaliser plusieurs itérations du modèle. Or, certains algorithmes d'assimilation de données néces-

2. Voir <http://www.era-clim.eu/> pour une description détaillée du projet

3. Réanalyse couplée effectuée par le Centre Européen de Prévisions Météorologiques à Moyen Terme dans le cadre du projet ERA-Clim2. Voir <https://www.ecmwf.int/en/forecasts/datasets/reanalysis-datasets/cera-20c> pour plus d'informations.

4. International workshop on coupled data assimilation, à Toulouse (2016), <http://www.meteo.fr/cic/meetings/2016/CDAW2016/index.html>

sitent également de réaliser plusieurs itérations du modèle afin de produire une analyse. Nous voyons alors là une opportunité pour « renforcer » le couplage des modèles lors de l'assimilation de données couplée. En effet, nous pourrions tirer parti des itérations de minimisation pour coupler les différents modèles à la manière des méthodes de couplage itératives. Nous tenterons dans ce manuscrit, divisé en trois parties, de répondre à ce problème en proposant différentes méthodes d'assimilation couplée.

- La première partie a pour objectif d'introduire les notions essentielles à la compréhension du problème de l'assimilation de données couplée. Nous commencerons par présenter les méthodes usuelles d'assimilation puis le problème du couplage océan-atmosphère dans les modèles de prévisions. Enfin nous introduirons les solutions proposées depuis les années 1990 pour assimiler des données de manière couplée.
- La seconde partie de ce manuscrit introduit quelques algorithmes d'assimilation de données couplée que nous avons pu développer au cours de ces travaux de thèse. Nous proposerons alors une étude de convergence de ces algorithmes avant de réaliser quelques premiers tests sur un modèle linéaire.
- Enfin, nous développerons un modèle couplé non-linéaire rendant mieux compte de la complexité du couplage océan-atmosphère. Puis nous incorporerons ce modèle dans une surcouche logicielle pour y assimiler des données de manière couplée. Nous évaluerons alors la robustesse des algorithmes introduits grâce à ce modèle.

Première partie

ÉTAT DES LIEUX

Cette première partie du manuscrit a pour but d'aborder le problème de l'assimilation de données couplée. S'agissant d'un domaine de recherche à cheval entre deux disciplines, l'assimilation de données et le couplage de modèles, nous prendrons le temps de présenter ces deux sujets de manière suffisamment détaillée avant de s'intéresser à l'état de l'art de l'assimilation de données couplée. De plus, chacune des deux communautés de recherche utilisant des notations et formalismes différents, nous introduirons petit à petit dans cette partie des notations unifiées qui seront utilisées dans la suite du manuscrit. L'ensemble de ces notations sera finalement résumé à la fin de cette partie.

- Nous commencerons par présenter des méthodes usuelles d'assimilation de données (non couplée) dans le but de comprendre le fonctionnement de chacune d'entre elles. Cette compréhension des méthodes est essentielle pour l'étude et le développement de nouveaux algorithmes d'assimilation de données couplée. En effet, ces nouveaux algorithmes utilisent les méthodes usuelles d'assimilation qu'il nous faut donc bien maîtriser.
 - L'assimilation de données couplée ne se limite cependant pas à l'utilisation des seules méthodes usuelles, mais fait aussi intervenir le couplage entre les modèles de prévision. C'est pourquoi le deuxième chapitre de cette partie sera dédié au problème du couplage de modèles. L'étude du couplage entre les modèles nous permettra alors d'apprécier la qualité du couplage dans les méthodes d'assimilation de données couplée présentées.
 - Nous pourrions finalement, dans le chapitre 3, étudier les différentes méthodes d'assimilation de données couplée qui ont été introduites depuis les années 1990. Nous essaierons d'apporter un regard critique sur l'ensemble de ces méthodes afin de pouvoir cibler du mieux possible nos objectifs.
-

Chapitre 1

Méthodes usuelles d'assimilation de données

Sommaire

1.1 Principe de l'assimilation de données	11
1.1.1 Discrétisation d'un système de prévision	13
1.1.2 Quelques définitions	14
1.1.3 Catégorisation des méthodes	16
1.2 Les méthodes stochastiques	17
1.2.1 Best Linear Unbiased Estimator et Interpolation Optimale	17
1.2.2 Le Filtre de Kalman	18
1.2.3 Limitation et difficulté majeure pour implémenter un filtre de Kalman	19
1.2.4 Filtre de Kalman d'Ensemble (EnKF)	19
1.3 Les méthodes variationnelles	21
1.3.1 Définition de la fonction coût dans un cadre stationnaire	21
1.3.2 4D-Var	23
1.3.3 4D-Var Incremental	26
1.3.4 Difficultés récurrentes en assimilation de données variationnelle . . .	28

Nous allons dans ce premier chapitre présenter différentes méthodes d'assimilation de données. Nous commencerons par définir concrètement ce qu'est l'assimilation de données en introduisant le vocabulaire adapté puis en définissant deux grandes classes de méthodes : les méthodes stochastiques et les méthodes variationnelles. Nous décrirons alors plusieurs méthodes usuelles que nous retrouverons dans la suite de ce manuscrit.

1.1 Principe de l'assimilation de données

La qualité des prévisions d'un modèle numérique dépend fortement de la qualité de l'analyse fournie à ce dernier pour définir aussi bien son état initial que ses conditions aux frontières. Cette analyse est créée par l'assimilation de données qui combine diverses informations disponibles en considérant leurs incertitudes respectives.

Pour mieux comprendre ce qu'est l'assimilation de données, prenons un exemple simple de la prévision de l'état x d'un système au cours du temps. Un modèle \mathcal{M} permet de représenter l'évolution temporelle de x . On se place à l'instant $t = 0$, la prévision de l'état du système correspond alors à la trajectoire :

$$x(t) = \mathcal{M}_{0 \rightarrow t}(x_0)$$

avec $t \in [0, T]$ la fenêtre temporelle considérée et $x_0 = x(0)$ l'état initial du système. L'assimilation de données a alors pour but d'évaluer le meilleur état initial x_0 possible pour le modèle \mathcal{M} , c'est à dire le plus proche de l'état initial réel du système. Notons que des méthodes d'assimilation de données permettent également d'estimer des paramètres du modèle pour améliorer la qualité des prévisions mais nous ne nous y intéresserons pas. Classiquement, pour évaluer x_0 , nous avons à notre disposition deux principales sources d'informations :

- des observations y^o à différents instants
- une ébauche x_0^b à l'instant t_0 .

Prise en compte des observations

La première source d'information sont les observations y^o du système. Notons \mathcal{O} l'espace des observations et \mathcal{S} l'espace de la trajectoire $x(t)$ du modèle. y^o et $x(t)$ n'étant pas nécessairement dans le même espace, il est nécessaire de recourir à un opérateur d'observations $\mathcal{H} : \mathcal{O} \rightarrow \mathcal{S}$ pour projeter les observations dans l'espace du modèle. Une première approche de l'assimilation de données serait alors :

$$\text{Trouver } x_0 \text{ tel que } y^o = \mathcal{H}(\mathcal{M}_{0 \rightarrow t}(x_0)) \quad (1.1)$$

Cette seule source d'information ne suffit cependant pas puisque :

- les observations sont généralement entachées d'erreurs qui ne sont pas connues
- le problème est souvent sous-déterminé car le nombre d'observations est généralement bien inférieur au nombre d'inconnues du problème. Dans ce cas le problème inverse (1.1) peut admettre une infinité de solutions

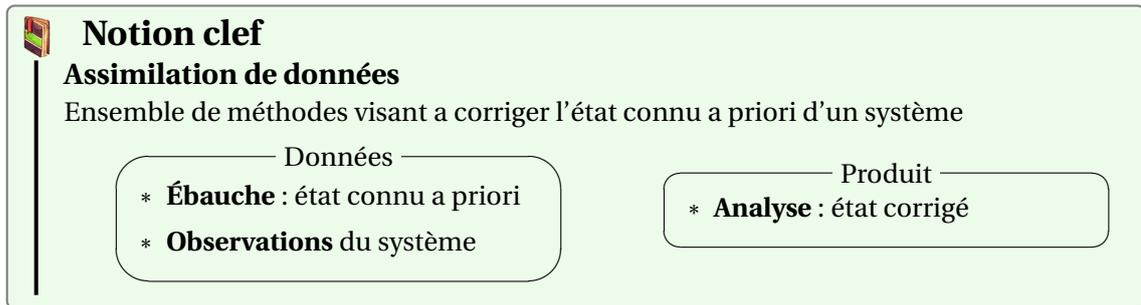
C'est pourquoi il est nécessaire d'utiliser une seconde source d'information, l'ébauche x_0^b .

Prise en compte de l'ébauche

La deuxième source d'information sur le système est un premier estimé de l'état du système : l'ébauche x_0^b . On cherchera alors à ce que la solution x_0 , en plus d'être proche des observations, soit proche de l'ébauche x_0^b en fonction de son incertitude par rapport à celles des observations. Le problème de l'assimilation de données devient alors :

$$\text{Trouver } x_0 \text{ tel que : } \begin{cases} \mathcal{H}(\mathcal{M}_{0 \rightarrow t}(x_0)) & \text{soit proche de } y^o \\ x_0 & \text{soit proche de } x_0^b \end{cases}$$

L'ébauche est généralement fournie par une prévision antérieure ce qui permet de contraindre la solution x_0 vers un état cohérent avec les équations du modèle considéré. De plus, l'ébauche contient également des informations sur les observations assimilées dans le passé. Sa prise en compte permet finalement de sur-déterminer le problème.



Avant d'aller plus loin dans la présentation des méthodes d'assimilation de données, nous allons tout d'abord introduire la discrétisation temporelle et spatiale du modèle \mathcal{M} et de sa trajectoire $x(t)$.

1.1.1 Discrétisation d'un système de prévision

Un modèle \mathcal{M} est un opérateur continu constitué d'un ensemble d'équations complexes décrivant la dynamique d'un système considéré (atmosphère, océan...). Le but d'une prévision est d'estimer la trajectoire $x(t)$ qui représente la solution des équations du modèle. Or, la complexité de ces équations implique qu'il est en général impossible de trouver une expression analytique de cette trajectoire. Une solution pour accéder à cette trajectoire est alors de discrétiser le modèle \mathcal{M} . Des schémas de discrétisation en temps et en espace permettent alors de résoudre les équations du modèle de manière approchée. Nous allons dans cette partie introduire les notations dites *discrètes* qui seront utilisées dans la suite de ce manuscrit.

Discrétisation en temps

Considérons que le modèle \mathcal{M} soit intégré sur une fenêtre temporelle $[0, T]$. La discrétisation en temps consiste à décomposer cette fenêtre en $N + 1$ pas de temps ($N \in \mathbb{N}$) tel que $T = 0 + N \cdot \Delta t$ où Δt représente la résolution temporelle. Nous noterons alors les pas de temps t_i tels que $t_i = i \cdot \Delta t$ où $i \in [0, N]$ sera désigné comme l'indice temporel. L'intégration du modèle $\mathcal{M}_{0 \rightarrow T}$ est alors discrétisée en une suite de N intégrations : $\mathcal{M}_{t_i \rightarrow t_{i+1}}$ avec $i \in [0, N - 1]$.

Discrétisation en espace

Pour pouvoir résoudre numériquement les équations du modèle, il faut également les discrétiser en espace. Nous noterons M le modèle discrétisé de \mathcal{M} . En lien avec la discrétisation temporelle, nous noterons \mathbf{x}_i le vecteur discrétisé spatialement de x à l'instant t_i : $x(t_i)$. L'état du système $x(t)$ au cours du temps est alors discrétisé en $N + 1$ états \mathbf{x}_i avec $i \in [0, N]$ évalués grâce au modèle discret M . Une intégration du modèle $\mathcal{M}_{t_i \rightarrow t_{i+1}}$ sera alors notée en version discrétisée : $M_{i,i+1}$ ou M_{i+1} . Nous pouvons écrire à partir de ces notations que :

$$\mathbf{x}_i = M_{0,i}(\mathbf{x}_0) = M_i(M_{i-1}(\dots M_1(\mathbf{x}_0))\dots)$$

où \mathbf{x}_0 est l'état initial discrétisé de x_0 . Le tableau 1.1 récapitule ces notations discrètes en lien avec les notations continues. Dans la suite de ce chapitre, nous ferons le choix de représenter les modèles et trajectoires uniquement sous forme discrète par souci de clarté.

Description	Notation continue	Notation discrète
Fenêtre d'intégration du modèle	$[0, T]$	$[0, T]$
Indice de pas de temps	-	i
Nombre de pas de temps	-	N
Durée d'un pas de temps	-	Δt
Temps	t	$t_i, \forall i \in \llbracket 0, N \rrbracket$
Modèle	\mathcal{M}	M
États du système	$x(t), \forall t \in [0, T]$	$\mathbf{x}_i, \forall i \in \llbracket 0, N \rrbracket$
Intégration du modèle jusqu'au temps t_i	$x(t_i) = \mathcal{M}_{0 \rightarrow t_i}(x_0)$	$\mathbf{x}_i = M_{0,i}(\mathbf{x}_0)$
Espace de l'état du système	\mathcal{S}	S
Espace des observations	\mathcal{O}	O

Tableau 1.1 – Tableau introduisant les notations discrétisées d'un modèle et de sa trajectoire au cours du temps

1.1.2 Quelques définitions

À présent que la discrétisation des modèles a été introduite, nous allons donner ici quelques définitions qui nous seront utiles pour présenter les différentes méthodes d'assimilation de données. Notons que les notations utilisées dans la suite seront conformes aux notations unifiées pour l'assimilation de données proposées par Ide *et al.* (1997). Voici donc quelques définitions importantes relatives à l'assimilation de données :

Définition 1.1.1 - Vecteur d'état

On appelle vecteur d'état l'ensemble des variables qui permettent de définir la trajectoire d'un modèle. On le note $\mathbf{x} \in S$.

Définition 1.1.2 - État vrai

L'état vrai d'un système est l'état réel dans lequel il se trouve, que l'on cherchera à estimer le plus précisément possible. Il est noté $\mathbf{x}^t \in S$ (t pour *true*) et est généralement inconnu.

Définition 1.1.3 - Ébauche

L'ébauche est l'état connu du système a priori. On la note $\mathbf{x}^b \in S$ (b pour *background*) et $\boldsymbol{\varepsilon}^b = \mathbf{x}^b - \mathbf{x}^t$ est l'erreur d'ébauche par rapport à l'état vrai, associée à une matrice de covariance d'erreur \mathbf{B} .

Définition 1.1.4 - Analyse

L'analyse notée $\mathbf{x}^a \in S$ (a pour *analysis*) est l'état du système estimé en sortie du processus d'assimilation de données. L'analyse est censée être la plus proche possible de l'état vrai mais est cependant entachée d'une erreur $\boldsymbol{\varepsilon}^a = \mathbf{x}^a - \mathbf{x}^t$ associée à une matrice de covariance d'erreur \mathbf{A} .

Définition 1.1.5 - Observations

Le vecteur d'observations du système à un instant t_i est noté $\mathbf{y}_i^o \in O$. Les observations sont également entachées d'erreurs $\boldsymbol{\varepsilon}_i^o = \mathbf{y}_i^o - H_i \mathbf{x}_i^t$ associées à une matrice de covariance \mathbf{R}_i (où \mathbf{x}_i^t représente l'état vrai du système au temps t_i et H_i est l'opérateur d'observations défini ci-après).

Définition 1.1.6 - Opérateur d'observations

On définit $H_i : O \rightarrow S$ l'opérateur d'observations permettant de projeter les observations dans l'espace du système au temps t_i .

Définition 1.1.7 - Modèle

On note $M : S \rightarrow S$ le modèle permettant de décrire l'évolution de l'état du système au cours du temps. Rappelons que nous avons $\mathbf{x}_i = M_{0,i}\mathbf{x}_0$ avec \mathbf{x}_0 l'état initial du système. Le modèle est également entaché d'une erreur $\boldsymbol{\eta}_i = M_{i-1,i}\mathbf{x}_{i-1} - \mathbf{x}_i^t$ associée à une matrice de covariance d'erreur \mathbf{Q}_i au temps t_i .

Définition 1.1.8 - Opérateurs linéaire tangent et adjoint

Dans la plupart des cas, le modèle M présentera des non-linéarités. Il sera alors souvent nécessaire d'avoir à disposition l'opérateur *linéaire tangent* associé, défini comme la jacobienne de M :

$$\mathbf{M}_{i,i+1} = \left. \frac{\partial M_{i,i+1}}{\partial \mathbf{x}} \right|_{\mathbf{x}_i}$$

Nous pouvons ensuite définir l'opérateur *adjoint* au modèle linéaire tangent : $\mathbf{M}_{i+1,i}^T$

Définition 1.1.9 - Opérateurs d'observations linéaire tangent et adjoint

L'opérateur d'observations tangent linéaire \mathbf{H}_i et son adjoint associé \mathbf{H}_i^T sont définis à partir de l'opérateur H_i de manière analogue aux modèles linéaire tangent et adjoint.

Nous pouvons déjà noter que l'état vrai \mathbf{x}^t étant inconnu, il est alors complexe d'obtenir les différentes matrices de covariances d'erreurs \mathbf{B} , \mathbf{A} , \mathbf{Q}_i , \mathbf{R}_i . De nombreuses méthodes existent dans la littérature proposant diverses estimations de ces matrices (Bannister, 2008a,b). Le tableau 1.2 récapitule l'ensemble des notations introduites ci-dessus.

Nom	Vecteur	Erreur	Matrice de covariance d'erreur	Opérateur	Opérateur linéaire	Opérateur adjoint
État vrai	\mathbf{x}^t	-	-	-	-	-
État du système	\mathbf{x}_i	$\boldsymbol{\eta}_i = M_{i-1,i}\mathbf{x}_{i-1} - \mathbf{x}_i^t$	\mathbf{Q}_i	M	\mathbf{M}	\mathbf{M}^T
Ébauche	\mathbf{x}^b	$\boldsymbol{\epsilon}^b = \mathbf{x}^b - \mathbf{x}^t$	\mathbf{B}	-	-	-
Analyse	\mathbf{x}^a	$\boldsymbol{\epsilon}^a = \mathbf{x}^a - \mathbf{x}^t$	\mathbf{A}	-	-	-
Observations	\mathbf{y}_i^o	$\boldsymbol{\epsilon}_i^o = \mathbf{y}_i^o - H_i(\mathbf{x}_i^t)$	\mathbf{R}_i	H_i	\mathbf{H}_i	\mathbf{H}_i^T

Tableau 1.2 – Tableau récapitulatif des notations utilisées dans les méthodes d'assimilation de données

Dans la suite, pour présenter les différentes méthodes d'assimilation de données, nous ferons le choix de nous placer dans un cadre d'étude relativement simple :

- $S = \mathbb{R}^n$ avec $n \in \mathbb{N}$
- $O = \mathbb{R}^p$ avec $p \in \mathbb{N}$
- $M : \mathbb{R}^n \rightarrow \mathbb{R}^n$
- $\mathbf{M}, \mathbf{M}^T \in \mathbb{R}^{n,n}$
- $\mathbf{x}, \mathbf{x}^a, \mathbf{x}^b \in \mathbb{R}^n$
- $\mathbf{B}, \mathbf{A} \in \mathbb{R}^{n,n}$
- $H : \mathbb{R}^n \rightarrow \mathbb{R}^p$
- $\mathbf{H} \in \mathbb{R}^{p,n}, \mathbf{H}^T \in \mathbb{R}^{n,p}$
- $\mathbf{y}^o \in \mathbb{R}^p$
- $\mathbf{R} \in \mathbb{R}^{p,p}$

où $\mathbb{R}^{a,b}$ désigne l'ensemble des matrices à coefficients réels avec a lignes et b colonnes.

1.1.3 Catégorisation des méthodes

Le nudging

Le nudging est sûrement la plus simple et intuitive des méthodes d'assimilation de données. Le principe est d'effectuer une correction de l'ébauche \mathbf{x}_i^b au temps t_i à partir de son écart aux observations et d'une matrice de gain $\mathbf{K} \in \mathbb{R}^{n,p}$:

Analyse par nudging

$$\mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{K}(\mathbf{y}_i^o - H_i(\mathbf{x}_i^b)) \quad (1.2)$$

Classiquement nous avons $\mathbf{K} = \alpha \mathbf{H}^T$ avec $\alpha \in \mathbb{R}$ évalué de manière empirique. $(\mathbf{y}_i^o - H_i(\mathbf{x}_i^b))$ est le terme de rappel qui permet de corriger l'ébauche à l'instant i à partir des observations. Cette méthode permet de définir simplement l'analyse \mathbf{x}_i^a au cours du temps et a pu faire ses preuves dans les années 90 en océanographie et météorologie. Des résultats de convergence du nudging existent mais n'englobent pas les problèmes géophysiques, le temps d'intégration des modèles étant trop court. Une alternative est alors d'utiliser une méthode de nudging direct et rétrograde (Auroux et Blum, 2005). Le principe est d'effectuer une étape du nudging classique (de t_0 à t_i), puis de réaliser un second nudging avec le modèle intégré de manière rétrograde (de t_i à t_0). Le processus est ensuite réitéré jusqu'à convergence. La convergence n'est cependant pas assurée, notamment dans le cas des problèmes diffusifs. Une extension a alors pu être proposée pour ces problèmes, appelée nudging direct et rétrograde diffusif (Auroux *et al.*, 2013). De manière générale, il est possible de se référer par exemple à Lakshmivarahan et Lewis (2013) pour plus de détails concernant les méthodes de nudging.

Méthodes stochastiques et variationnelles

Nous pouvons regrouper les autres méthodes d'assimilation de données en deux grandes catégories : les méthodes **stochastiques** et les méthodes **variationnelles** que nous allons définir.

Définition 1.1.10 - Méthodes stochastiques

Les méthodes **stochastiques** sont des méthodes statistiques qui estiment l'état du système le plus probable à un moment donné au regard des observations disponibles, compte tenu du modèle et de l'évaluation de leurs erreurs respectives.

Définition 1.1.11 - Méthodes variationnelles

L'assimilation de données **variationnelle** est basée sur la théorie du contrôle optimal. Le principe est de minimiser une fonction coût qui mesure les erreurs d'un modèle par rapport à différentes sources d'information sur une fenêtre temporelle donnée. Ces erreurs représentent généralement les écarts aux observations et à l'ébauche et sont pondérées grâce à des matrices de covariance.

Il existe différentes méthodes au sein de chacune de ces deux catégories qui auront leurs avantages et inconvénients. Il convient alors d'évaluer dans chaque cas, pour chaque application, quelle sera la méthode la plus performante, la moins coûteuse, etc. La suite de ce chapitre est

consacrée à la description de plusieurs algorithmes d'assimilation de données que nous seront amenés à rencontrer dans la littérature de l'assimilation de données couplée (chapitre 3).

1.2 Les méthodes stochastiques

Les méthodes stochastiques sont des méthodes dont le principe est de corriger l'ébauche grâce aux observations du système comme le propose le nudging. La différence est que les méthodes stochastiques cherchent à évaluer la meilleure matrice \mathbf{K} possible pour évaluer l'état d'analyse \mathbf{x}^a optimal compte tenu des observations, de l'ébauche, du modèle et des hypothèses faites sur les erreurs de ces informations. Nous allons introduire petit à petit ces méthodes stochastiques en commençant par présenter une méthode un peu plus complexe que le nudging, le *Best Linear Unbiased Estimator* (BLUE) ainsi qu'une implémentation numérique réaliste du BLUE, l'Interpolation Optimale (OI). Nous présenterons enfin les filtres de Kalman.

1.2.1 Best Linear Unbiased Estimator et Interpolation Optimale

Le BLUE consiste à estimer l'état vrai \mathbf{x}_i^t du système à un temps t_i à partir d'une combinaison linéaire de l'ébauche \mathbf{x}_i^b et des observations \mathbf{y}_i^o comme le fait le nudging. La différence est alors de proposer une estimation optimale de la matrice \mathbf{K} en fonction des erreurs d'ébauches et d'observations. Les hypothèses du BLUE sont les suivantes :

Hypothèses BLUE

- Les observations et l'ébauche sont non biaisées
- L'opérateur d'observations est linéaire
- Les matrices de covariance d'erreurs sont connues
- Les erreurs d'ébauche et d'observations sont décorrélées
- $\mathbb{E}(\boldsymbol{\varepsilon}^b) = 0$ — $\mathbb{E}(\boldsymbol{\varepsilon}_i^o) = 0$
- $\mathbb{E}(\boldsymbol{\varepsilon}^b \boldsymbol{\varepsilon}^{bT}) = \mathbf{B}$
- $\mathbf{y}_i^o = \mathbf{H}_i \mathbf{x}_i^t + \boldsymbol{\varepsilon}_i^o$ — $\mathbb{E}(\boldsymbol{\varepsilon}_i^o \boldsymbol{\varepsilon}_i^{oT}) = \mathbf{R}_i$

Le problème est le suivant : quelles sont les matrices $\boldsymbol{\beta}$ et \mathbf{K} permettant d'obtenir la meilleure estimation possible \mathbf{x}_i^a telle que $\mathbf{x}_i^a = \boldsymbol{\beta} \mathbf{x}_i^b + \mathbf{K} \mathbf{y}_i^o$? Le BLUE propose une solution en recherchant une estimation non biaisée ayant la plus petite variance possible :

$$\left\{ \begin{array}{l} \mathbb{E}(\boldsymbol{\varepsilon}^a) = 0 \\ tr(\mathbf{A}) \text{ minimum} \end{array} \right.$$

ces conditions sont satisfaites lorsque :

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

$$\mathbf{A} = (\mathbf{I} - \mathbf{K}_\star \mathbf{H}_i) \mathbf{B}$$

on peut se référer par exemple à Daget (2007) pour un détail des calculs. Nous obtenons ainsi les équations du BLUE :

Équations BLUE

$$\begin{aligned}
 \mathbf{K}_\star &= \mathbf{B}\mathbf{H}_i^T(\mathbf{H}_i\mathbf{B}\mathbf{H}_i^T + \mathbf{R}_i)^{-1} && \text{Matrice de gain optimal} \\
 \mathbf{x}_i^a &= \mathbf{x}_i^b + \mathbf{K}_\star(\mathbf{y}_i^o - \mathbf{H}_i\mathbf{x}_i^b) && \text{Etape d'analyse} \\
 \mathbf{A} &= (\mathbf{I} - \mathbf{K}_\star\mathbf{H}_i)\mathbf{B} && \text{Matrice de covariance d'erreur a-posteriori}
 \end{aligned} \tag{1.3}$$

Les principales limitations de cette méthode sont d'une part ses hypothèses assez restrictives (opérateur d'observation linéaire) et d'autre part la possible très grande dimension de la matrice \mathbf{K}_\star , qui rend la méthode inutilisable en pratique. L'Interpolation Optimale (OI pour Optimal Interpolation en anglais) propose un moyen de contourner ce problème de grande dimension en simplifiant le calcul de \mathbf{K}_\star . Cela consiste à résoudre les équations du BLUE (1.3) pour chaque variable du problème et non plus dans sa totalité. Le but est alors de réduire considérablement la taille du problème en ne considérant qu'un certain nombre d'observations au voisinage de chaque point de grille. On résout ainsi une somme de petits problèmes plus simples que le problème initial, avec un coût de calcul raisonnable. Cependant, l'analyse produite est souvent bruitée du fait de l'utilisation d'observations différentes pour chaque point de grille.

1.2.2 Le Filtre de Kalman

Le filtre de Kalman (KF pour *Kalman Filter*) a été introduit par Kalman (1960) puis Kalman et Bucy (1961). Le but de ce filtre est de corriger à chaque pas de temps t_i l'état du système à partir des informations disponibles, par exemple les observations y_i^o . La correction se base sur la solution du BLUE introduit précédemment.

Hypothèses du filtre de Kalman

- A l'instant initial, \mathbf{x}_0^b est entaché d'une erreur $\boldsymbol{\varepsilon}_0^b$, associée à une matrice de covariance \mathbf{B}_0
- Le modèle est linéaire, ses erreurs sont indépendantes et suivent une loi Gaussienne : $\mathbf{x}_{i+1} = \mathbf{M}_{i,i+1}\mathbf{x}_i + \boldsymbol{\eta}_i$, avec $\boldsymbol{\eta}_i \sim \mathcal{N}(0, \mathbf{Q}_i)$ et $\mathbb{E}(\boldsymbol{\eta}_{i_1}\boldsymbol{\eta}_{i_2}^T) = 0$ si $i_1 \neq i_2$
- L'opérateur d'observations est linéaire : $\mathbf{y}_i^o = \mathbf{H}_i\mathbf{x}_i + \boldsymbol{\varepsilon}_i^o$
- Les erreurs d'observations sont indépendantes et suivent une loi Gaussienne : $\boldsymbol{\varepsilon}_i^o \sim \mathcal{N}(0, \mathbf{R}_i)$ et $\mathbb{E}(\boldsymbol{\varepsilon}_{i_1}^o\boldsymbol{\varepsilon}_{i_2}^o{}^T) = 0$ si $i_1 \neq i_2$
- Les erreurs de différents types sont indépendantes : $\mathbb{E}(\boldsymbol{\varepsilon}_i^o\boldsymbol{\varepsilon}_0^b{}^T) = 0$, $\mathbb{E}(\boldsymbol{\eta}_{i_1}\boldsymbol{\varepsilon}_{i_2}^o{}^T) = 0$, $\mathbb{E}(\boldsymbol{\eta}_i\boldsymbol{\varepsilon}_0^b{}^T) = 0$

Le principe du KF est d'itérer pour chaque pas de temps :

- une étape d'**analyse** identique au BLUE pour corriger l'ébauche
- une étape de **prévision** à partir de l'analyse jusqu'au pas de temps suivant.

La particularité de ce filtre est que l'on a accès directement à une estimation des matrices de covariances \mathbf{B} et \mathbf{A} qui évoluent grâce à la dynamique du modèle et au réseau d'observations.

Filtre de Kalman**Algorithme 1.1** Filtre de Kalman**Initialisation** \mathbf{x}_0^b et \mathbf{B}_0 **Étape d'analyse** $\mathbf{K}_i = \mathbf{B}_i \mathbf{H}_i^T [\mathbf{H}_i \mathbf{B}_i \mathbf{H}_i^T + \mathbf{R}_i]^{-1}$ $\mathbf{x}_i^a = \mathbf{x}_i^b + \mathbf{K}_i (\mathbf{y}_i^o - \mathbf{H}_i \mathbf{x}_i^b)$ $\mathbf{A}_i = (\mathbf{I} - \mathbf{K}_i \mathbf{H}_i) \mathbf{B}_i$ **Étape de prévision** $\mathbf{x}_{i+1}^b = \mathbf{M}_{i,i+1} \mathbf{x}_i^a$ $\mathbf{B}_{i+1} = \mathbf{M}_{i,i+1} \mathbf{A}_i \mathbf{M}_{i,i+1}^T + \mathbf{Q}_i$ **1.2.3 Limitation et difficulté majeure pour implémenter un filtre de Kalman**

Une des principales limitations de ce filtre est son hypothèse selon laquelle le modèle et l'opérateur d'observation doivent être linéaires. C'est pourquoi une version étendue de ce filtre a été proposée pour répondre aux problèmes non linéaires (Jazwinski, 1970) que l'on appelle Filtre de Kalman Étendu (EKF pour Extended Kalman Filter en anglais) mais qui en pratique ne fonctionne pas bien. En effet, l'EKF est confronté à la même difficulté que le KF, à savoir la dimension du problème. Concrètement, les modèles météorologiques ou océanographiques impliquent des millions de variables, la matrice de covariance d'erreur \mathbf{B}_i peut donc être de l'ordre de $\mathbb{R}^{10^9, 10^9}$, voire plus, et l'on cherche constamment à augmenter cet ordre. Le stockage et la manipulation d'une telle matrice \mathbf{B}_i est impossible numériquement. Il est possible de procéder à une diminution du rang de la matrice \mathbf{B} par une analyse en composantes principales par exemple (Cane *et al.*, 1996), mais les applications réalistes se détournent plutôt de ces méthodes pour préférer l'utilisation des filtres de Kalman d'ensemble que nous allons présenter.

1.2.4 Filtre de Kalman d'Ensemble (EnKF)

Pour contourner les difficultés liées au calcul des matrices de covariances d'erreur, Evensen (1997) a proposé une version ensembliste du filtre de Kalman : le Filtre de Kalman d'Ensemble (EnKF pour *Ensemble Kalman Filter* en anglais). Le principe de l'EnKF est de réduire la taille du problème en estimant la trajectoire du modèle à partir d'un ensemble de trajectoires et d'observations. Ces ensembles sont générés en appliquant de petites perturbations autour de l'ébauche et des observations à partir des erreurs supposées du modèle et des erreurs d'observations. Notons N_m le nombre de membres de notre ensemble, et m l'indice du membre dans l'ensemble ($m \in [1, N_m]$).

Hypothèses EnKf

- M est le modèle dynamique non linéaire
- \mathbf{M} est le modèle tangent linéaire de M
- H est l'opérateur d'observations non linéaire
- \mathbf{H} est l'opérateur linéaire d'observation tangent de H
- Chaque membre m de l'ensemble est généré à partir de distributions gaussiennes. On note pour chaque membre, $\forall m \in [1, N_m]$:
 - $\boldsymbol{\varepsilon}_{i,m}^o$ les erreurs d'observations au temps t_i telles que $\boldsymbol{\varepsilon}_{i,m}^o \sim \mathcal{N}(0, \mathbf{R}_i)$
 - $\boldsymbol{\eta}_{i,m}$ les erreurs modèles au temps t_i telles que $\boldsymbol{\eta}_{i,m} \sim \mathcal{N}(0, \mathbf{Q}_i)$

Les différents membres $m \in [1, N_m]$ de l'ensemble pour chaque pas de temps $i \in [0, N]$ seront donc générés à partir des erreurs $\boldsymbol{\varepsilon}_{i,m}^o$ et $\boldsymbol{\eta}_{i,m}$ suivant des lois gaussiennes. Une version possible de l'EnKF est alors la suivante :

Filtre de Kalman d'Ensemble (EnKF)

Algorithme 1.2 Filtre de Kalman d'Ensemble

Initialisation	Générer $\forall m \in [1, N_m]$: $\boldsymbol{\eta}_{0,m} \sim \mathcal{N}(0, \mathbf{Q}_0)$ $\mathbf{x}_{0,m}^b = \mathbf{x}_0^b + \boldsymbol{\eta}_{0,m}$
Étape d'analyse	$\bar{\mathbf{x}}_i^b = \frac{1}{N_m} \sum_{m=1}^{N_m} \mathbf{x}_{i,m}^b$ $\mathbf{B}_i \simeq \frac{1}{N_m - 1} \sum_{m=1}^{N_m} (\mathbf{x}_{i,m}^b - \bar{\mathbf{x}}_i^b) (\mathbf{x}_{i,m}^b - \bar{\mathbf{x}}_i^b)^T$ $\mathbf{H}_i \mathbf{B}_i \simeq \frac{1}{N_m - 1} \sum_{m=1}^{N_m} (H_i(\mathbf{x}_{i,m}^b) - \overline{H_i(\mathbf{x}_i^b)}) (\mathbf{x}_{i,m}^b - \bar{\mathbf{x}}_i^b)^T$ $\mathbf{H}_i \mathbf{B}_i \mathbf{H}_i^T \simeq \frac{1}{N_m - 1} \sum_{m=1}^{N_m} (H_i(\mathbf{x}_{i,m}^b) - \overline{H_i(\mathbf{x}_i^b)}) (H_i(\mathbf{x}_{i,m}^b) - \overline{H_i(\mathbf{x}_i^b)})^T$ $\mathbf{K}_i = (\mathbf{H}_i \mathbf{B}_i)^T [\mathbf{H}_i \mathbf{B}_i \mathbf{H}_i^T + \mathbf{R}_i]^{-1}$
Étape de prévision	Générer $\forall m \in [1, N_m]$: $\boldsymbol{\varepsilon}_{i,m}^o \sim \mathcal{N}(0, \mathbf{R}_i)$ $\mathbf{y}_{i,m}^o = \mathbf{y}_i^o + \boldsymbol{\varepsilon}_{i,m}^o$ $\mathbf{x}_{i,m}^a = \mathbf{x}_{i,m}^b + \mathbf{K}_i (\mathbf{y}_{i,m}^o - H_i(\mathbf{x}_{i,m}^b))$
	(1.4)

Le problème de stockage posé par le KF et l'EKf est alors évité. En effet, les seules informations

à transmettre entre les étapes de prévision et d'analyse sont des ensembles de N_m vecteurs d'états $\mathbf{x}_{i,m}^{a,b}$. Ainsi, le coût le plus élevé de cette méthode est l'inversion de la matrice d'innovation $(\mathbf{H}_i \mathbf{B}_i \mathbf{H}_i^T + \mathbf{R}_i)$ de taille $p \times p$ où p est le nombre d'observations. Malheureusement cela est également infaisable en pratique étant donné que le nombre d'observations est souvent de l'ordre de 10^4 , 10^5 . Une stratégie pour résoudre ce problème est par exemple de ne considérer qu'un sous-ensemble d'observations pour chaque point d'analyse, de manière similaire à l'OI, comme cela est proposé par Ott *et al.* (2004) avec l'algorithme LEKF (Local EnKF). Le gain de Kalman \mathbf{K}_i est alors différent pour chaque point de grille mais l'inversion est possible numériquement. Notons qu'il existe de nombreux autres filtres de Kalman dérivant de cette méthode d'ensemble que nous décrirons succinctement lorsque nous serons amenés à les rencontrer dans la suite de ce manuscrit.



Méthodes clés

Assimilation de données stochastique

- **Interpolation optimale (OI)** : Implémentation du BLUE (1.3) où les observations sont sous-échantillonnées pour chaque point d'analyse
 - + Coût de calcul raisonnable
 - Analyse bruitée
- **Filtre de Kalman (KF)** : Correction en temps réel de l'état d'un système
 - + Accès direct à \mathbf{B} grâce à la dynamique du modèle
 - Hypothèse restrictive : le modèle doit être linéaire
 - Problème de grande dimension : \mathbf{B}_i de l'ordre de $\mathbb{R}^{10^9, 10^9}$
- **Filtre de Kalman d'ensemble (EnKF)** : Utilisation de prévisions d'ensemble pour estimer \mathbf{B}
 - + Modèle non-linéaire et dimension du problème réduite
 - Coût de calcul élevé

1.3 Les méthodes variationnelles

Les méthodes variationnelles sont basées sur la théorie du contrôle optimal, cherchant à estimer \mathbf{x}^a tel que :

$$\mathbf{x}^a = \underset{\mathbf{x}}{\operatorname{arg\,min}} J(\mathbf{x}) \quad (1.5)$$

où le vecteur d'état du système \mathbf{x} est appelé vecteur de contrôle et J est une fonction coût à minimiser. J permet de mesurer les écarts aux observations et à l'ébauche par exemple. Il ne s'agit donc plus de méthodes statistiques à proprement parler, même si les statistiques sont nécessaires pour définir la fonction coût à minimiser.

1.3.1 Définition de la fonction coût dans un cadre stationnaire

Une interprétation classique du problème (1.5) est de considérer l'analyse comme étant l'état le plus probable au regard du modèle et des observations. Nous définissons alors la fonction coût à minimiser à partir de considérations statistiques. C'est à dire que l'analyse correspond au maximum de la fonction de densité de probabilité de \mathbf{x}_0 sachant $\mathbf{y}_0^o : p(\mathbf{x}_0 | \mathbf{y}_0^o)$. D'après la théorie de Bayes, nous avons que :

Théorème 1.3.1 - Théorème de Bayes

$$p(\mathbf{x}_0 | \mathbf{y}_0^o) = \frac{p(\mathbf{y}_0^o | \mathbf{x}_0) p(\mathbf{x}_0)}{p(\mathbf{y}_0^o)}$$

Le problème de l'assimilation de données revient donc à :

$$\text{Trouver } \mathbf{x}_0^a \text{ tel que } \mathbf{x}_0^a = \arg \max_{\mathbf{x}_0} p(\mathbf{x}_0 | \mathbf{y}_0^o) = \arg \max_{\mathbf{x}_0} \frac{p(\mathbf{y}_0^o | \mathbf{x}_0) p(\mathbf{x}_0)}{p(\mathbf{y}_0^o)} \quad (1.6)$$

La densité de probabilité $p(\mathbf{y}_0^o)$ ne dépendant pas de \mathbf{x}_0 le problème (1.6) est équivalent à :

$$(1.6) \Leftrightarrow \mathbf{x}_0^a = \arg \max_{\mathbf{x}_0} p(\mathbf{y}_0^o | \mathbf{x}_0) p(\mathbf{x}_0) \quad (1.7)$$

Sous l'hypothèse que $\boldsymbol{\varepsilon}^o \sim \mathcal{N}(\mathbf{H}\mathbf{x}_0, \mathbf{R}^{-1})$ et $\boldsymbol{\varepsilon}^b \sim \mathcal{N}(\mathbf{x}_0, \mathbf{B}^{-1})$, il est possible de définir les deux densités de probabilité qui entrent en jeu dans ce problème à partir des matrices de covariance d'erreur, en supposant qu'elles suivent une loi gaussienne :

$$\begin{aligned} p(\mathbf{y}_0^o | \mathbf{x}_0) &= p(H_0(\mathbf{x}_0) = \mathbf{y}_0^o) \\ &= L_y \exp \left[-\frac{1}{2} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o)^T \mathbf{R}^{-1} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o) \right] \end{aligned}$$

$$\begin{aligned} p(\mathbf{x}_0) &= p(\mathbf{x}_0 = \mathbf{x}_0^b) \\ &= L_b \exp \left[-\frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) \right] \end{aligned}$$

où L_y et L_b sont deux constantes positives indépendantes de \mathbf{x}_0 . Le problème de maximisation devient donc équivalent à un problème de minimisation :

$$\begin{aligned} (1.7) &\Leftrightarrow \arg \max_{\mathbf{x}_0} \left(\exp \left[-\frac{1}{2} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o)^T \mathbf{R}^{-1} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o) - \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) \right] \right) \\ &\Leftrightarrow \arg \min_{\mathbf{x}_0} \left(\frac{1}{2} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o)^T \mathbf{R}^{-1} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o) + \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) \right) \end{aligned}$$

Le problème initial (1.6) est finalement un problème de minimisation de la fonction coût J telle que :

$$J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o)^T \mathbf{R}^{-1} (H_0(\mathbf{x}_0) - \mathbf{y}_0^o) \quad (1.8)$$

Ainsi, en minimisant cette fonction coût, on minimise les écarts de l'état \mathbf{x}_0 avec l'ébauche \mathbf{x}_0^b et les observations \mathbf{y}_0^o . Cette fonction coût stationnaire est celle qui est minimisée dans l'algorithme d'assimilation de données dit du *3D-Var* (Lorenç, 1986) (*3D* car le vecteur de contrôle \mathbf{x}_0 peut se trouver dans les trois dimensions de l'espace et *Var* car il s'agit d'une méthode variationnelle).

Équivalence avec le BLUE

Dans le cas où l'opérateur d'observation H est linéaire, le problème introduit précédemment est équivalent à minimiser la fonction coût suivante :

$$J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} (\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0)^T \mathbf{R}^{-1} (\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0)$$

Comme $J(\mathbf{x}_0)$ est quadratique en \mathbf{x}_0 , J est convexe et possède un minimum qui est atteint lorsque son gradient ∇J est nul. Le problème de minimisation se ramène donc à estimer \mathbf{x}_0 tel que $\nabla J(\mathbf{x}_0) = 0$:

$$\begin{aligned} \nabla J(\mathbf{x}_0) &= \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0) \\ \nabla J(\mathbf{x}_0^a) &= 0 \\ \Leftrightarrow \mathbf{B}^{-1}(\mathbf{x}_0^a - \mathbf{x}_0^b) - \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0^a) &= 0 \\ \Leftrightarrow \mathbf{x}_0^a &= \mathbf{x}_0^b + (\mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1}(\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0^b) \end{aligned}$$

On peut ensuite montrer que $(\mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H})^{-1} \mathbf{H}^T \mathbf{R}^{-1} = \mathbf{B} \mathbf{H}^T (\mathbf{H} \mathbf{B} \mathbf{H}^T + \mathbf{R})^{-1}$ (formule de Sherman-Morrison-Woodbury) et on retrouve alors la formulation exacte du BLUE à l'optimum (équation (1.3)) : $\mathbf{x}_0^a = \mathbf{x}_0^b + \mathbf{K}^* (\mathbf{y}_0^o - \mathbf{H}\mathbf{x}_0^b)$. Le problème de minimisation de la fonction coût (1.8) est donc équivalent au BLUE lorsque l'opérateur d'observation H est linéaire.

Objectif des méthodes variationnelles

Le problème de minimisation pouvant être équivalent à celui du BLUE, nous pourrions calculer directement le gain optimal \mathbf{K}^* pour évaluer l'état du système. Cependant nous avons pu observer que calculer ce gain n'est pas envisageable du fait des trop grandes dimensions du problème. L'objectif des méthodes variationnelles est donc de chercher le minimum de J sans calculer \mathbf{K}^* , ce qui peut être fait grâce à un algorithme de descente tel que l'algorithme du gradient conjugué. Il s'agit d'un processus itératif qui consiste à minimiser $J(\mathbf{x})$ en évaluant son gradient $\nabla J(\mathbf{x})$ pour mettre à jour \mathbf{x} dans la direction du gradient. L'évaluation de $\nabla J(\mathbf{x})$ peut se faire via une méthode adjointe (Le Dimet et Talagrand, 1986).

Informations de la hessienne de J

On définit la hessienne de J comme étant $\nabla(\nabla J) = \nabla^2 J$. La hessienne peut être utile pour évaluer la qualité de l'analyse à convergence. En effet, dans un cadre linéaire, nous avons que $\nabla^2 J = \mathbf{B}^{-1} + \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H}$ et on peut alors démontrer que :

$$\mathbf{A} = (\nabla^2 J)^{-1}$$

De plus, le conditionnement de $\nabla^2 J$ impactera directement la vitesse de convergence de la minimisation. Le conditionnement d'une matrice est le rapport entre sa plus grande et sa plus petite valeur propre, que nous appellerons nombre de conditionnement. Un mauvais conditionnement de $\nabla^2 J$, caractérisé par un nombre de conditionnement élevé, impliquera que ∇J est faiblement convexe et donc une minimisation très lente. En revanche, un bon conditionnement de $\nabla^2 J$, caractérisé par un nombre de conditionnement faible, sera le signe d'une forte convexité de ∇J , impliquant une convergence de la minimisation plus rapide.

1.3.2 4D-Var

La méthode dite du 4D-Var nous permet d'introduire la dynamique temporelle dans le problème de l'assimilation de données variationnelle. On se place sur une fenêtre temporelle $[0, T]$ sur laquelle le système évolue. Le principe du 4D-Var est alors d'évaluer l'état du système sur cette fenêtre temporelle au regard de l'ébauche \mathbf{x}_0^b , des observations \mathbf{y}_i^o et de la dynamique du modèle. Puisque l'état du système sur la fenêtre temporelle est directement lié à sa condition

initiale, le problème du 4D-Var se ramène à l'évaluation de \mathbf{x}_0^a . La figure 1.1 représente schématiquement le fonctionnement du 4D-Var.

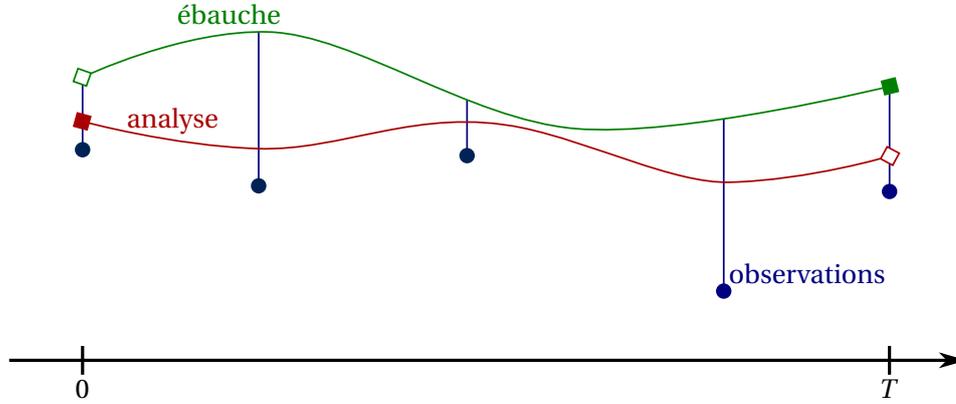


FIGURE 1.1 – Principe du 4D-Var : on se place sur une fenêtre temporelle $[0, T]$ sur laquelle nous disposons d'une trajectoire ébauche et d'observations ponctuelles. Le 4D-Var évalue la meilleure trajectoire possible en fonction de l'ébauche et des observations : l'analyse.

Hypothèses 4D-Var

- Le système est dynamique et évolue sur une fenêtre temporelle avec un modèle M parfait
- Des observations sont disponibles sur l'ensemble de la fenêtre temporelle
- Le vecteur de contrôle \mathbf{x}_0 correspond à l'état du modèle à $t = 0$

Les observations permettent de faire intervenir la dynamique du modèle. En effet, le vecteur de contrôle étant \mathbf{x}_0 , l'intégration du modèle permet d'évaluer la trajectoire $\mathbf{x}_i = M_{0,i}\mathbf{x}_0, \forall i \in [0, N]$ qui sera comparée aux observations. La fonction coût à minimiser est alors définie ainsi :

$$\begin{aligned}
 J(\mathbf{x}_0) &= J^b(\mathbf{x}_0) + J^o(\mathbf{x}_0) \\
 &= \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N (\mathbf{y}_i^o - H_i(\mathbf{x}_i))^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(\mathbf{x}_i)) \\
 &= \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N ((\mathbf{y}_i^o - H_i(M_{0 \rightarrow i}(\mathbf{x}_0)))^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(M_{0 \rightarrow i}(\mathbf{x}_0)))) \quad (1.9) \\
 &= \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \\
 &\quad \frac{1}{2} \sum_{i=0}^N (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))
 \end{aligned}$$

où on appellera J^b le terme de la fonction coût relevant de l'ébauche et J^o la partie relative aux observations. On peut ensuite exprimer le gradient de $J(\mathbf{x}_0)$:

$$\begin{aligned}
 \nabla J(\mathbf{x}_0) &= \nabla J^b(\mathbf{x}_0) + \nabla J^o(\mathbf{x}_0) \\
 &= \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - \sum_{i=0}^N \mathbf{M}_1^T \dots \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))
 \end{aligned}$$

où $\mathbf{M}_1^T \dots \mathbf{M}_i^T$ fait référence aux intégrations successives du modèle adjoint $\mathbf{M}_{i \rightarrow 0}^T$.

Fonction coût et gradient du 4D-Var

$$J(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))$$

$$\nabla J(\mathbf{x}_0) = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - \sum_{i=0}^N \mathbf{M}_1^T \dots \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} (\mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))$$

Calcul pratique du gradient

Le calcul du gradient $\nabla J(\mathbf{x}_0)$ n'est pas chose aisée. Une méthode classique consiste à considérer $d_i = \mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0)))))$ le vecteur d'écart aux observations puis factoriser $\nabla J^o(\mathbf{x}_0)$ ainsi :

$$-\nabla J^o(\mathbf{x}_0) = \mathbf{H}_0^T \mathbf{R}_0^{-1} d_0 + \mathbf{M}_1^T [\mathbf{H}_1^T \mathbf{R}_1^{-1} d_1 + \mathbf{M}_2^T [\mathbf{H}_2^T \mathbf{R}_2^{-1} d_2 + \dots + \mathbf{M}_N^T \mathbf{H}_N^T \mathbf{R}_N^{-1} d_N]]$$

Cette factorisation permet de calculer $\nabla J^o(\mathbf{x}_0)$ grâce à :

- une intégration du modèle direct pour évaluer l'ensemble des innovations d_i
- une intégration en mode *reverse* (c'est à dire de t_N à t_0) du modèle adjoint pour évaluer $\nabla J^o(\mathbf{x}_0)$. Cette intégration à l'envers du modèle adjoint est nécessaire pour évaluer $\mathbf{M}_N^T \mathbf{H}_N^T \mathbf{R}_N^{-1} d_N$ avant de calculer $\mathbf{M}_{N-1}^T [\mathbf{H}_{N-1}^T \mathbf{R}_{N-1}^{-1} d_{N-1} + \mathbf{M}_N^T \mathbf{H}_N^T \mathbf{R}_N^{-1} d_N]$ etc... pour finalement calculer $\nabla J^o(\mathbf{x}_0)$.

L'algorithme suivant présente la méthode 4D-Var où un critère d'arrêt ε et un nombre maximal d'itérations l_{\max} sont définis.

Algorithme du 4D-Var

Algorithme 1.3 4D-Var

Initialisation : $\mathbf{x}_0 = \mathbf{x}_0^b$, $l = 0$

while $\frac{\|\nabla J(\mathbf{x}_0)\|}{\|\nabla J(\mathbf{x}_0^b)\|} > \varepsilon$ ou $l < l_{\max}$ **do**

Calcul de J via le calcul des innovations grâce au modèle M :

$$d_i = \mathbf{y}_i^o - H_i(M_i(M_{i-1}(\dots(M_1(\mathbf{x}_0))))))$$

$$J(\mathbf{x}_0) = \frac{1}{2}(\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N d_i^T \mathbf{R}_i^{-1} d_i$$

Calcul de $\nabla J(\mathbf{x}_0)$ grâce aux modèle et opérateur adjoints \mathbf{M}^T et \mathbf{H}^T en mode *reverse* :

$$\nabla J(\mathbf{x}_0) = \mathbf{B}^{-1}(\mathbf{x}_0 - \mathbf{x}_0^b) - [\mathbf{H}_0^T \mathbf{R}_0^{-1} d_0 + \mathbf{M}_1^T [\mathbf{H}_1^T \mathbf{R}_1^{-1} d_1 + \mathbf{M}_2^T [\mathbf{H}_2^T \mathbf{R}_2^{-1} d_2 + \dots + \mathbf{M}_N^T \mathbf{H}_N^T \mathbf{R}_N^{-1} d_N]]]$$

Descente et mise à jour de \mathbf{x}_0

$l = l + 1$

end while

L'avantage du 4D-Var est qu'il permet d'ajouter la dynamique du modèle comme information dans l'assimilation de données ce qui en fait sa force au regard de méthodes telles que le 3D-Var cité précédemment. Cependant, les non-linéarités d'un modèle rendront la fonction coût à minimiser non-quadratique, ce qui rendra la minimisation de cette dernière moins efficace. Afin de répondre à ce problème, Courtier *et al.* (1994) ont introduit une variante du 4D-Var : le 4D-Var incrémental.

1.3.3 4D-Var Incrémental

La méthode dite du 4D-Var incrémental introduite par Courtier *et al.* (1994) pour les modèles de prévision propose une solution du 4D-Var par itérations successives avec une fonction coût quadratique dans le cas où des non-linéarités sont présentes dans le modèle. Cette méthode est par exemple utilisée au CEPMMT depuis 1997 et à Météo-France depuis 2000. L'idée est d'écrire le problème sous forme d'incrément en définissant $\delta \mathbf{x}_0 = \mathbf{x}_0 - \mathbf{x}_0^b$. On exprime alors $M_i(\mathbf{x}_0)$ et $H_i(\mathbf{x}_i)$ grâce au développement de Taylor :

$$\begin{aligned} M_i(\mathbf{x}_0) &= M_i(\mathbf{x}_0^b + \delta \mathbf{x}_0) = M_i(\mathbf{x}_0^b) + \mathbf{M}_i \delta \mathbf{x}_0 + \mathcal{O}(\delta \mathbf{x}_0) \\ H_i(\mathbf{x}_i) &= H_i(\mathbf{x}_i^b + \delta \mathbf{x}_i) = H_i(\mathbf{x}_i^b) + \mathbf{H}_i \delta \mathbf{x}_i + \mathcal{O}(\delta \mathbf{x}_i) \end{aligned}$$

et l'on suppose que $\delta \mathbf{x}_0$ est « petit ». Le problème de minimisation peut donc être reformulé en fonction de l'incrément $\delta \mathbf{x}_0$, la fonction coût devenant linéaire :

$$\begin{aligned} J(\delta \mathbf{x}_0) &= J^b(\delta \mathbf{x}_0) + J^o(\delta \mathbf{x}_0) \\ J^b(\delta \mathbf{x}_0) &= \frac{1}{2} \delta \mathbf{x}_0^T \mathbf{B}^{-1} \delta \mathbf{x}_0 \\ J^o(\delta \mathbf{x}_0) &= \frac{1}{2} \sum_{i=0}^N (d_i - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0)^T \mathbf{R}_i^{-1} (d_i - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0) \end{aligned} \tag{1.10}$$

où les d_i correspondent aux innovations préalablement calculées avec le modèle non-linéaire :

$$d_i = \mathbf{y}_i^o - H_i(M_i(\dots(M_1(\mathbf{x}_0^b))))$$

La fonction coût (1.10) est alors une approximation quadratique de la fonction coût du 4D-Var (1.9). L'algorithme du 4D-Var incrémental est constitué de deux boucles :

- une boucle externe calculant les innovations d_i grâce au modèle non linéaire,
- une boucle interne minimisant la fonction coût quadratique incrémentale (1.10) pour estimer l'incrément d'analyse $\delta \mathbf{x}_0^a$.

L'incrément $\delta \mathbf{x}_0^a$ permet ensuite de calculer un nouvel état d'analyse à partir de l'ébauche, puis l'on réitère ce processus si la solution n'est pas satisfaisante. L'algorithme du 4D-Var incrémental peut alors être vu comme un algorithme de Gauss-Newton comme nous le verrons dans le chapitre 4. Notons $\cdot^{(p)}$ les itérations de boucles externes, nous proposons de résumer ci-dessous l'expression de la fonction coût, du gradient et l'algorithme du 4D-Var incrémental en fonction des itérations de boucles externes :

Fonction coût et gradient du 4D-Var incrémental

$$\begin{aligned}
\mathbf{x}_0^{(0)} &= \mathbf{x}_0^b \\
\delta \mathbf{x}_0^{(p)} &= \mathbf{x}_0^{(p)} - \mathbf{x}_0^{(p-1)} \\
d_i^{(p)} &= \mathbf{y}_i^o - H_i(M_i(\dots(M_1(\mathbf{x}_0^{(p)})))) \\
J(\delta \mathbf{x}_0^{(p)}) &= \frac{1}{2} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right)^T \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) + \\
&\quad \frac{1}{2} \sum_{i=0}^N \left(d_i^{(p)} - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)} \right)^T \mathbf{R}_i^{-1} \left(d_i^{(p)} - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)} \right) \\
\nabla J(\delta \mathbf{x}_0^{(p)}) &= \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) - \sum_{i=0}^N \mathbf{M}_1^T \dots \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} (d_i - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)})
\end{aligned}$$

Algorithme du 4D-Var incrémental**Algorithme 1.4** 4D-Var incrémental**Initialisation :** $\mathbf{x}_0^{(0)} = \mathbf{x}_0^b$

BOUCLE EXTERNE

Intégration du modèle non-linéaire : $\mathbf{x}_i^{(p)} = M_{0 \rightarrow i}(\mathbf{x}_0^{(p)})$ Calcul des innovations $d_i^{(p)} = \mathbf{y}_i^o - H_i \mathbf{x}_i^{(p)}$

BOUCLE INTERNE

Calcul de $J(\delta \mathbf{x}_0^{(p)})$ grâce aux opérateurs linéarisés \mathbf{M} et \mathbf{H} Calcul du gradient $\nabla J(\delta \mathbf{x}_0^{(p)})$ grâce aux opérateurs adjoints linéaires \mathbf{M}^T et \mathbf{H}^T Minimisation : évaluation de l'incrément d'analyse $\delta \mathbf{x}_0^{a,(p)}$

FIN DE LA BOUCLE INTERNE

Mise à jour de l'état de référence $\mathbf{x}_0^{(p+1)} = \mathbf{x}_0^{(p)} + \delta \mathbf{x}_0^{a,(p)}$

FIN DE LA BOUCLE EXTERNE

Calcul de l'état d'analyse $\mathbf{x}_0^a = \mathbf{x}_0^{(p+1)}$ et $\mathbf{x}_i^a = M_{0 \rightarrow i}(\mathbf{x}_0^a)$

L'avantage majeur de cette méthode est que la fonction coût J à minimiser est quadratique. Cela améliore grandement les performances de la minimisation par rapport à un algorithme où J ne serait pas quadratique en raison des non-linéarités du modèle. On peut trouver dans l'article de Gratton *et al.* (2007) quelques résultats théoriques concernant la convergence des méthodes incrémentales. Nous pouvons également mentionner des études pratiques de convergence dans des conditions particulières comme celle de Trémolet (2007). Ces études montrent que la convergence du 4D-Var incrémental n'est pas toujours assurée, notamment dans le cas de systèmes complexes où des approximations du modèle sont effectuées pour simplifier le problème.

1.3.4 Difficultés récurrentes en assimilation de données variationnelle

La taille du problème peut engendrer des difficultés de calcul et/ou de stockage des données. En effet ces méthodes variationnelles nécessitent de stocker l'intégralité de la trajectoire du modèle pour calculer les innovations d_i et d'effectuer deux intégrations du modèle : une directe et une adjointe. Deux autres points principaux peuvent poser problème. Le premier concerne les difficultés à modéliser la matrice \mathbf{B} , essentielle au bon fonctionnement des algorithmes et le second portera sur le développement des opérateurs adjoints.

Modélisation de la matrice \mathbf{B}

La prise en compte de l'ébauche permet au problème d'être bien posé et peut alors être vu comme un terme de régularisation de Tikhonov. La matrice \mathbf{B} a donc de nombreux rôles au sein de l'assimilation de données comme cela peut être décrit dans l'article de Bannister (2008a) :

- elle quantifie la précision de l'ébauche par définition même de cette matrice,
- elle diffuse l'information l'observée dans l'espace. Par exemple dans le cas du 3D-Var, la solution est donnée par :

$$\mathbf{x}_0^a = \mathbf{x}_0^b + \underbrace{\mathbf{B}}_{\in \mathcal{S} \times \mathcal{S}} \underbrace{\mathbf{H}^T}_{\in \mathcal{S} \times \mathcal{O}} \underbrace{(\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1}}_{\in \mathcal{O}} (\mathbf{y}^o - \mathbf{H}\mathbf{x}_0^b)$$

On remarque alors que la matrice \mathbf{B} permet de diffuser l'information provenant des observations : $\mathbf{H}^T (\mathbf{H}\mathbf{B}\mathbf{H}^T + \mathbf{R})^{-1} (\mathbf{y}^o - \mathbf{H}\mathbf{x}_0^b)$. Dans le cadre du 4D-Var, l'opérateur \mathbf{H} est remplacé par \mathbf{HM} , et le modèle aide alors également à cette diffusion, mais cela ne suffit généralement pas car la fenêtre temporelle d'assimilation est limitée. L'apport d'informations de la matrice \mathbf{B} reste donc essentielle,

- elle permet de proposer un état d'analyse cohérent en imposant un équilibre physique entre les variables. Cela est réalisé grâce aux termes de covariance croisés (qui ne sont pas sur la diagonale principale) de \mathbf{B} .

C'est pourquoi il est important de pouvoir modéliser correctement cette matrice pour pouvoir proposer le meilleur état d'analyse possible. Cependant la dimension de \mathbf{B} étant trop grande en pratique pour être représentée directement (de l'ordre de $10^9 \times 10^9$ par exemple), il est nécessaire de la modéliser, ce qui est à l'origine de la complexité du problème. Weaver et Courtier (2001) proposent par exemple de modéliser les corrélations d'erreur à partir d'une équation de diffusion générale. La bonne représentation de la matrice d'erreur d'ébauche reste un domaine de recherche très actif de nos jours. Cependant, nos travaux ne portant pas sur la modélisation de \mathbf{B} en elle même, nous nous contenterons de considérer une matrice \mathbf{B} simple (diagonale ou multi-diagonale) dans des problèmes de petite dimension permettant sa représentation directe.

Utilisation des opérateurs adjoints pour les méthodes variationnelles

Une seconde difficulté du problème d'assimilation de données dans le cadre des méthodes du 4D-Var et 4D-Var incrémental est l'utilisation des opérateurs adjoints et notamment du modèle adjoint. En effet, le développement d'un modèle adjoint peut-être complexe pour un modèle de prévision opérationnel et sa qualité aura un impact majeur sur la qualité des analyses produites. De plus, l'utilisation d'un modèle adjoint nécessite le développement numérique de l'adjoint et de sa maintenance, puisqu'un modèle de prévision évolue en permanence. Le coût

de développement d'un modèle adjoint est donc relativement élevé dans les centres de prévisions et n'est pas à sous évaluer. C'est pourquoi des méthodes dites variationnelles d'ensemble commencent à se développer (Liu *et al.*, 2008, 2009), permettant de se passer du modèle adjoint, mais nous ne discuterons pas de ces méthodes dans ce manuscrit. Nous pouvons enfin rappeler que l'utilisation du modèle adjoint nécessite de stocker la trajectoire du modèle, ce qui peut-être également coûteux.



Méthodes clefs

Assimilation de données variationnelle

- **4D-Var** : Estimation de la trajectoire sur une fenêtre temporelle par minimisation d'une fonction coût
 - + La dynamique du modèle permet une analyse cohérente et un impact des observations sur toute la fenêtre
 - Minimisation difficile en présence de fortes non-linéarités
- **4D-Var incrémental** : Algorithme itératif proposant à convergence une solution identique au 4D-Var
 - + Minimisation de fonctions coût quadratiques
 - ↳ Les non-linéarités du modèle n'impactent pas la minimisation quadratique

Difficultés principales :

- Modélisation de **B** complexe
- Développement d'un modèle adjoint nécessaire

Chapitre 2

Le couplage au sein des modèles de prévision

Sommaire

2.1 Modélisation et physique de l'océan et de l'atmosphère	32
2.1.1 Description des milieux	32
2.1.2 Modélisation de l'océan et de l'atmosphère	33
2.1.3 Expression des flux turbulents	35
2.2 Le problème du couplage océan-atmosphère	37
2.3 Méthodes de couplage usuelles et méthode de Schwarz	38
2.3.1 Vocabulaire du couplage	38
2.3.2 Couplage synchrone au pas de temps	39
2.3.3 Couplage asynchrone par fenêtres temporelles	39
2.3.4 Couplage par méthode itérative de Schwarz	40

Dans l'idéal, un modèle de prévision du système terrestre doit faire interagir différentes composantes entre elles. En effet, l'atmosphère par exemple est en contact direct avec l'océan et ces deux milieux échangent des flux en permanence, il en va de même avec la glace qui interagit avec l'océan et l'atmosphère. Pour simplifier, la figure 2.1 présente schématiquement les interactions entre les principaux milieux d'un système terrestre. Un modèle de prévision le plus réaliste et juste possible prendrait en compte toutes ces interactions entre ces différents milieux. Cependant, créer un modèle unique comprenant toutes les composantes du système terrestre, et les faisant interagir entre elles en son sein est inenvisageable pour des raisons techniques, de coût, etc. Une solution est alors de modéliser chaque composante par un modèle différent, et de coupler ces modèles via des *coupleurs* qui gèrent les échanges à leurs interfaces, tel que OASIS-MCT (Valcke *et al.*, 2015). Le problème du couplage entre les différents modèles de prévision (atmosphère, océan, terre, glace...) est aujourd'hui un facteur clé pour la modélisation du climat et pour la qualité des prévisions météorologiques. A titre d'exemple, il a pu être mis en évidence que la qualité du couplage océan-atmosphère pouvait grandement influencer la prévision météorologique de l'intensité des cyclones tropicaux ainsi que la prévision de leur trajectoire (voir par exemple Lemarié *et al.* (2014)). De manière plus générale, une bonne représentation de ce couplage permet une meilleure représentation des phénomènes météorologiques tropicaux, une meilleure prévisibilité du phénomène d'El Niño (Cane *et al.*, 1986)

ou encore des prévisions climatiques à long et moyen termes plus justes. Le couplage océan-atmosphère prend donc une place de plus en plus importante dans la communauté scientifique pour le développement de nouveaux modèles de prévisions et de climat.

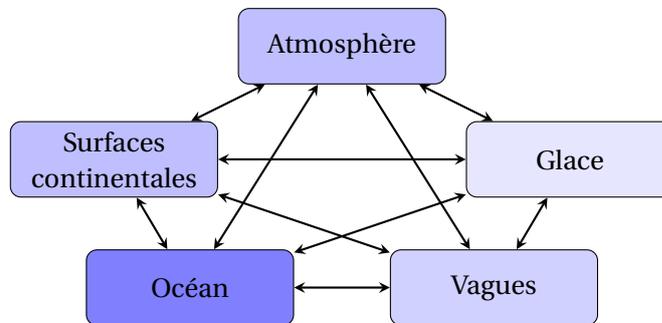


FIGURE 2.1 – Différentes interactions entre des milieux du système terrestre. Chaque système interagit avec l'ensemble des autres par l'intermédiaire d'échanges de flux de chaleur, de tensions de vent, etc

Dans la suite, nous ne nous intéresserons qu'au seul couplage entre les modèles d'océan et d'atmosphère, mais cette présentation peut-être étendue à d'autres systèmes couplés via leur interface. Nous commencerons par présenter le problème du couplage océan-atmosphère puis quelques méthodes pour la résolution numérique de ce problème. Nous mettrons enfin l'accent sur une méthode itérative dite méthode de Schwarz, permettant d'obtenir une solution couplée au sens fort mathématiquement.

2.1 Modélisation et physique de l'océan et de l'atmosphère

Nous présentons dans cette partie les équations permettant de modéliser classiquement l'océan et l'atmosphère ainsi que leur comportement proche de l'interface air-mer. Nous simplifierons volontairement cette étude qui peut s'avérer plus complexe. En effet, la physique de l'océan et de l'atmosphère n'étant pas le cœur de notre problème, nous nous concentrerons sur l'essentiel afin d'en avoir une compréhension globale. Pour une étude plus détaillée on pourra se reporter au papier de Lemarié *et al.* (2015).

2.1.1 Description des milieux

Commençons par décrire la structure verticale des milieux d'atmosphère et d'océan. Nous considérerons l'atmosphère libre et l'océan intérieur loin de l'interface air-mer (voir 2.2). Ces milieux, par définition, ne sont pas influencés par la présence de l'interface océan-atmosphère, et ne seront donc pas étudiés en détail dans ce manuscrit. Nous avons ensuite quatre couches qui interagissent entre elles proche de l'interface, à savoir de haut en bas :

- la Couche Limite Atmosphérique (CLA),
- la Couche Limite de Surface Atmosphérique (CLSA),
- la Couche Limite de Surface Océanique (CLSO),
- la Couche Limite Océanique (CLO).

La CLSA et CLSO constituent ce que l'on appelle la couche limite de surface (CLS). Il s'agit d'une zone très turbulente où de nombreux échanges ont lieu entre l'atmosphère et l'océan. Dans la

suite, l'étude de ces flux turbulents dans la CLS nous permettra de comprendre toute la complexité du couplage océan-atmosphère. La CLA et la CLO sont deux couches plus éloignées de la surface mais également influencées par la turbulence présente proche de l'interface. Elles font en quelque sorte le lien entre l'interface, l'atmosphère libre et l'océan intérieur. La figure 2.2 représente schématiquement cette structure verticale et le tableau 2.1 donne quelques ordres de grandeur associés.

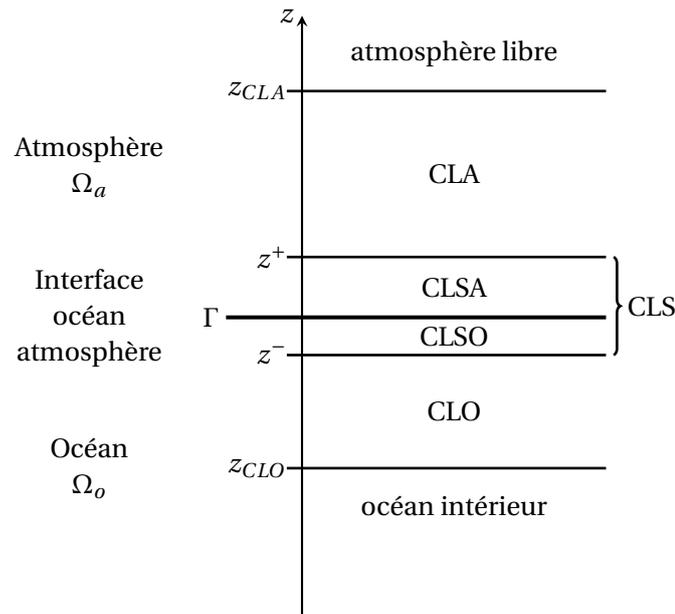


FIGURE 2.2 – Schématisation de la structure verticale de l'océan et atmosphère proche de l'interface. Nous noterons z_{CLA} la hauteur de la CLA, z_{CLO} la profondeur de la CLO et z^+ et z^- les limites de la CLS. Notons aussi que nous appellerons Γ l'interface entre l'océan et l'atmosphère.

Hauteur	Ordre de grandeur
z_{CLA}	$\sim 500 - 1000\text{m}$
z^+	$\sim 10 - 100\text{m}$
Profondeur	
z^-	$\sim 1 - 10\text{m}$
z_{CLO}	$\sim 50 - 100\text{m}$

Tableau 2.1 – Ordres de grandeur caractéristiques des couches limites océanique et atmosphérique

2.1.2 Modélisation de l'océan et de l'atmosphère

L'océan et l'atmosphère sont traités comme des fluides newtoniens dont l'évolution est décrite par les équations de Navier-Stokes, auxquelles s'ajoutent plusieurs hypothèses simplificatrices adaptées au cas de l'atmosphère ou de l'océan pour obtenir le système d'équations connu sous le nom d'équations primitives.

Hypothèses équations primitives

- Le fluide est en équilibre hydrostatique, ce qui consiste à négliger l'accélération verticale du fluide dans l'équation du mouvement vertical.
- Les termes impliquant la composante verticale de l'accélération de Coriolis sont négligés.
- Approximation de Boussinesq : on considère l'incompressibilité de l'eau. Nous considérons alors une densité de référence de l'eau ρ_0 à la place de ρ .
- Approximation de fluide parfait : on considère l'air comme un gaz parfait.
- Hypothèse de couche mince : le fluide est contenu dans une couche très mince relativement au rayon de la sphère. En pratique cela revient à négliger l'altitude par rapport au rayon de la terre.

Équations primitives (océan et atmosphère)

Équation de quantité de mouvement

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}_{3D} \cdot \nabla) \mathbf{u} = f_C \mathbf{k} \times \mathbf{u} - \frac{1}{\rho_0} \nabla_h p + \nu_m \Delta \mathbf{u} + f_{\mathbf{u}} \quad (2.1)$$

Équation d'état

$$\rho = \rho(p, \mathcal{T}) \quad (2.2)$$

Équilibre hydrostatique

$$\frac{\partial p}{\partial z} = -\rho g \quad (2.3)$$

Équation de continuité

$$\nabla \cdot \mathbf{u}_{3D} = 0 \quad (2.4)$$

Équation de conservation des traceurs

$$\frac{\partial \mathcal{T}}{\partial t} + \mathbf{u}_{3D} \cdot \nabla \mathcal{T} = \nu_s \Delta \mathcal{T} + f_{\mathcal{T}} \quad (2.5)$$

où ∇_h correspond à l'opérateur de gradient selon les deux dimensions horizontales de l'espace, et \mathbf{k} est l'axe de rotation de la terre. On note $\mathbf{u} = (u, v)^T$ les composantes zonale et méridienne de la vitesse de l'air ou de l'eau, et $\mathbf{u}_{3D} = (\mathbf{u}^T, w)^T$ le vecteur de courant 3D avec w la composante verticale de la vitesse. $\mathbf{u}_{3D} \cdot \nabla$ correspond alors à l'opérateur d'advection. f_C est le paramètre de Coriolis, g l'accélération de la pesanteur et ν_m et ν_s sont les coefficients de viscosité moléculaire et de diffusion moléculaire respectivement. p et ρ correspondent à la pression et à la masse volumique de l'air ou de l'eau et ρ_0 est la densité de référence du fluide. \mathcal{T} représente les différents traceurs considérés dans l'atmosphère et l'océan. Nous appelons traceurs des quantités scalaires transportées et mélangées par la dynamique du milieu. Dans notre cas, \mathcal{T} pourra représenter l'humidité de l'air q_a , la température potentielle de l'air ou de l'eau $\theta_{a,o}$ ou encore la salinité de l'eau S_o . Enfin les termes $f_{\mathbf{u}, \mathcal{T}}$ représentent les seconds membres de chacune des équations (termes source ou de rappel par exemple).

L'implémentation numérique des équations primitives (2.1), (2.2), (2.3), (2.4) et (2.5) ne permet cependant pas de résoudre l'ensemble des processus présents à proximité de la surface. En effet les dynamiques petites échelles présentes dans la CLS, liées au caractère très turbulent de l'air et de l'eau, ne peuvent être résolues du fait de la résolution de la grille des modèles trop grossière. Or ces flux turbulents ont un impact majeur sur la dynamique du fluide dans les couches limites. Il est donc nécessaire de quantifier ces turbulences dites *sous mailles* non résolues au sein de la CLS pour rendre compte correctement du couplage océan-atmosphère dans les modèles. Il existe de nombreux papiers sur l'étude des équations primitives, dont par exemple Lions *et al.* (1992a,b) ou encore McWilliams (2006).

2.1.3 Expression des flux turbulents

Pour comprendre comment s'expriment les flux turbulents dans la CLS, notons $\mathbf{x}(t) = (u, v, \mathcal{T})^T$ le vecteur d'état du modèle d'équations primitives (d'océan ou d'atmosphère) qui vérifie l'équation d'advection suivante :

$$\frac{\partial \mathbf{x}}{\partial t} + (\mathbf{u}_{3D} \cdot \nabla) \mathbf{x} = \tilde{f}_{\mathbf{x}} \quad (2.6)$$

où $\tilde{f}_{\mathbf{x}}$ représente tous les autres termes autre que l'advection dans les équations (2.1) et (2.5).

Décomposition de Reynolds

Pour résoudre les fluctuations turbulentes de \mathbf{x} présentes proche de l'interface air-mer, considérons que \mathbf{x} est composée d'échelles résolues et non résolues distinctes. Pour séparer ces échelles il existe diverses méthodes dans la littérature. Nous utiliserons la décomposition de Reynolds qui est une approche statistique et qui mène au système communément appelé RANS (Reynolds Average Navier-Stokes) :

$$\mathbf{x} = \langle \mathbf{x} \rangle + \mathbf{x}'$$

où $\langle \mathbf{x} \rangle$ correspond aux échelles *résolues* et \mathbf{x}' aux fluctuations sous-mailles non résolues par le modèle. Notons que l'opérateur statistique abstrait $\langle \cdot \rangle$ satisfait les propriétés de Reynolds :

$$\langle \mathbf{x}' \rangle = 0, \quad \langle \langle \mathbf{x} \rangle \rangle = \langle \mathbf{x} \rangle, \quad \left\langle \frac{\partial \mathbf{x}}{\partial s} \right\rangle = \frac{\partial \langle \mathbf{x} \rangle}{\partial s}, \quad \langle \mathbf{x}_\alpha + \mathbf{x}_\beta \rangle = \langle \mathbf{x}_\alpha \rangle + \langle \mathbf{x}_\beta \rangle$$

La décomposition de Reynolds appliquée à l'équation d'advection (2.6) vérifiée par \mathbf{x} donne alors :

$$\begin{aligned} & \langle \partial_t \langle \mathbf{x} \rangle + \partial_t \mathbf{x}' \rangle + \langle \langle (\mathbf{u}_{3D} \cdot \nabla) \mathbf{x} \rangle + (\mathbf{u}'_{3D} \cdot \nabla) \mathbf{x}' \rangle = \langle \tilde{f}_{\mathbf{x}} \rangle \\ \Leftrightarrow & \underbrace{\partial_t \langle \mathbf{x} \rangle + \nabla \cdot \langle (\mathbf{u}_{3D} \mathbf{x}) \rangle}_{(1)} + \underbrace{\nabla \cdot \langle (\mathbf{u}'_{3D} \mathbf{x}') \rangle}_{(2)} = \langle \tilde{f}_{\mathbf{x}} \rangle \end{aligned} \quad (2.7)$$

avec $\mathbf{u}_{3D} \mathbf{x} = (\mathbf{u}_{3D} u, \mathbf{u}_{3D} v, \mathbf{u}_{3D} \mathcal{T})^T$ (de même pour $\mathbf{u}'_{3D} \mathbf{x}'$). On remarque que l'équation d'advection se décompose à présent en deux termes : un terme (1) impliquant les échelles résolues par le modèle et un terme (2) faisant intervenir les fluctuations sous mailles non résolues.

Hypothèse de fermeture et paramétrisation

L'objectif est alors d'exprimer le terme $\langle \mathbf{u}'_{3D} \mathbf{x}' \rangle$ en fonction des grandeurs résolues de l'écoulement pour ainsi *fermer* le système, c'est à dire le rendre résoluble. Une hypothèse de fermeture classique du problème est la suivante (voir Stull (1988) pour une présentation plus complète des différents schémas de fermeture) :

$$\langle \mathbf{u}'_{3D} \mathbf{x}' \rangle = -(\mathbf{T}_x \nabla \langle \mathbf{x} \rangle)^T$$

où \mathbf{T}_x est une matrice diagonale composée des coefficients de turbulence. Dans le cadre du couplage océan-atmosphère, nous nous intéresserons particulièrement aux fluctuations verticales du flux $\partial_z \langle w' x' \rangle$ et négligerons les autres qui sont le plus souvent traitées comme des termes de viscosité ou diffusion classique. Nous obtenons :

$$(2.7) \Leftrightarrow \begin{aligned} \partial_t \langle \mathbf{x} \rangle + \nabla \cdot (\langle \mathbf{u}_{3D} \mathbf{x} \rangle) + \partial_z (\langle w' \mathbf{x}' \rangle) &= \langle \mathcal{F}_x \rangle \\ \Leftrightarrow \partial_t \langle \mathbf{x} \rangle + \nabla \cdot (\langle \mathbf{u}_{3D} \mathbf{x} \rangle) + \partial_z (K_x(z) \partial_z \langle \mathbf{x} \rangle) &= \langle \mathcal{F}_x \rangle \end{aligned}$$

où $K_x = \begin{pmatrix} K_m & 0 & 0 \\ 0 & K_m & 0 \\ 0 & 0 & K_s \end{pmatrix}$ correspond aux coefficients de \mathbf{T}_x relatifs à la verticale et :

$$K_x(z) \partial_z \langle \mathbf{x} \rangle = \begin{pmatrix} K_m(z) \partial_z u \\ K_m(z) \partial_z v \\ K_s(z) \partial_z \mathcal{T} \end{pmatrix}$$

avec $K_m(z)$ le coefficient de viscosité turbulente et $K_s(z)$ le coefficient de diffusivité turbulente. Le problème est à présent d'exprimer correctement $K_x(z)$. Cela est usuellement fait par des schémas de turbulence que l'on appelle des *paramétrisations* comme, entre autres, le schéma KPP (*K-Profile Parametrisation*) dans l'océan (Large *et al.*, 1994) ou un schéma TKE (*Turbulent Kinetic Energy*) dans l'atmosphère (Cuxart *et al.* (2000) par exemple). Notons que ces schémas de turbulence introduisent dans l'équation des non linéarités puisque $K_x(z)$ est souvent exprimé en fonction de la valeur du flux turbulent à l'interface $\langle w' \mathbf{x}' \rangle_0$ et de $\partial_z \langle \mathbf{x} \rangle$:

$$K_x(z) = f(\langle w' \mathbf{x}' \rangle_0, \partial_z \langle \mathbf{x} \rangle)$$

Flux turbulent à l'interface et formulations Bulk

Pour fermer intégralement le problème, il reste à définir la valeur du flux turbulent à l'interface $\langle w' \mathbf{x}' \rangle_0$. Des hypothèses simplificatrices permettent tout d'abord de considérer les flux constants au sein de la CLS, donc que $\langle w' \mathbf{x}' \rangle|_{CLS}(z) = \langle w' \mathbf{x}' \rangle_0$. À l'interface air-mer, $\langle w' \mathbf{x}' \rangle_0$ est communément exprimé via la formule suivante (Monin et Obukhov, 1954; Large, 2006a) :

$$\langle w' \mathbf{x}' \rangle_0 = -C_x |\Delta u| (\langle \mathbf{x}(z^+) \rangle - \langle \mathbf{x}(z^-) \rangle)$$

où $\Delta u = \langle u(z^+) \rangle - \langle u(z^-) \rangle$. Les coefficients C_x sont des fonctions non linéaires, souvent non différentiables, que l'on appelle *coefficients de transfert Bulk*. Ils permettent de lier les échelles turbulentes aux échelles résolues du problème. Les *formulations Bulk* permettent d'évaluer ces coefficients. Ce calcul n'est pas aisé et peut par exemple être réalisé avec un algorithme itératif de type recherche de point fixe (voir par exemple Pelletier *et al.* (2017) pour plus de détails).



Notion clef

Flux turbulents à l'interface océan-atmosphère

Dynamique petite échelle proche de l'interface air-mer liée au caractère turbulent de l'air et de l'eau

- La résolution typique des modèles de prévision rend impossible le calcul de la turbulence proche de l'interface
- La turbulence a un impact majeur sur l'évolution des fluides dans la CLS, CLA et CLO

Quelle solution pour la prendre en compte?

⇒ Plusieurs hypothèses permettent de se ramener au problème de l'expression de :

$$\begin{aligned} K_{\mathbf{x}}(z) & \quad \text{les coefficients de turbulence} \\ \langle w' \mathbf{x}' \rangle_0 & \quad \text{la valeur du flux à l'interface} \end{aligned}$$

- Des **schémas de turbulence** permettent d'exprimer $K_{\mathbf{x}}(z)$
 - Les **formulations Bulk** permettent d'exprimer $\langle w' \mathbf{x}' \rangle_0$
- ⇒ Ces schémas et formulations introduisent des non-linéarités et non-différentiabilités dans les modèles

2.2 Le problème du couplage océan-atmosphère

Maintenant que le comportement des milieux d'océan et d'atmosphère proche de l'interface a été présenté, nous pouvons nous pencher sur le problème du couplage océan-atmosphère en lui-même. Pour simplifier la présentation du problème, nous nous placerons dans le cas d'un modèle simple colonne. Nous noterons Ω_o et Ω_a les domaines d'océan et d'atmosphère comme introduits dans la figure 2.2 et l'interface air-mer sera notée Γ . Les bords extérieurs des deux domaines seront notés $\partial\Omega_o^{ext}$ et $\partial\Omega_a^{ext}$ et on désigne par \mathbf{x}_a et \mathbf{x}_o les vecteurs d'état atmosphériques et océaniques. Le problème du couplage est introduit par la consistance des flux turbulents requise à l'interface, c'est à dire que :

$$\begin{aligned} \rho_a \langle w'_a \mathbf{x}'_a \rangle_0 &= \rho_o \langle w'_o \mathbf{x}'_o \rangle_0 \\ \Leftrightarrow \rho_a K_{\mathbf{x}}^a \frac{\partial \langle \mathbf{x}_a \rangle}{\partial z} \Big|_{\Gamma} &= \rho_o K_{\mathbf{x}}^o \frac{\partial \langle \mathbf{x}_o \rangle}{\partial z} \Big|_{\Gamma} = \mathcal{F}_{oa}(\langle \mathbf{x}_{a|\Gamma} \rangle, \langle \mathbf{x}_{o|\Gamma} \rangle, \mathcal{R}) \end{aligned} \quad (2.8)$$

où ρ_a et ρ_o sont les densités de l'air et de l'eau et $K_{\mathbf{x}}^a$ et $K_{\mathbf{x}}^o$ sont les coefficients de turbulence de l'atmosphère et de l'océan. Nous désignons par $\mathbf{x}_{a|\Gamma}$ la restriction du champ \mathbf{x}_a à l'interface Γ . $\mathcal{F}_{oa}(\langle \mathbf{x}_{a|\Gamma} \rangle, \langle \mathbf{x}_{o|\Gamma} \rangle, \mathcal{R})$ permet alors de définir le flux océan-atmosphère à l'interface en fonctions de $\langle \mathbf{x}_{a|\Gamma} \rangle$, $\langle \mathbf{x}_{o|\Gamma} \rangle$ et \mathcal{R} des flux radiatifs s'ajoutant aux flux des traceurs. Dans la suite, nous nous affranchirons de la notation $\langle \cdot \rangle$ pour alléger l'écriture. Cette égalité des flux nous amène naturellement à poser le problème du couplage suivant :

$$\begin{aligned} & \text{Trouver } \mathbf{x}_a \text{ et } \mathbf{x}_o \text{ sur la fenêtre d'intégration } [0, T] \text{ tels que} \\ & \left\{ \begin{array}{ll} \mathcal{M}_a(\mathbf{x}_a) = f_a & \text{sur } \Omega_a \times [0, T] \\ \mathcal{M}_o(\mathbf{x}_o) = f_o & \text{sur } \Omega_o \times [0, T] \\ \mathcal{D}_a(\mathbf{x}_{a|\Gamma}) = \mathcal{D}_o(\mathbf{x}_{o|\Gamma}) = \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}, \mathbf{x}_{o|\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \end{array} \right. \quad (2.9) \end{aligned}$$

avec \mathcal{M}_a et \mathcal{M}_o les opérateurs différentiels définissant les modèles d'atmosphère et d'océan et f_a et f_o les termes sources des équations. \mathcal{D}_a et \mathcal{D}_o représentent les opérateurs d'interface permettant d'assurer la continuité des flux turbulents à l'interface air-mer (équation (2.8)). Le problème du couplage étant désormais posé, nous allons voir dans la suite différentes stratégies qui peuvent être adoptées pour approcher la solution de ce problème.

2.3 Méthodes de couplage usuelles et méthode de Schwarz

Nous pouvons tout d'abord distinguer deux catégories de méthodes, les méthodes de couplage dites *one-way* et *two-way*. Les premières imposent un forçage de l'océan sur l'atmosphère sans aucune rétroaction, le problème de couplage (2.9) n'est alors pas résolu. Les méthodes *two-way* permettent quant à elles une rétroaction en forçant l'océan par l'atmosphère en réponse au premier forçage. La solution produite par ces méthodes se rapproche ainsi plus de la solution du problème (2.9) mais ne le résout pas nécessairement. Pullen *et al.* (2006) ont pu montrer que des méthodes de couplage *two-way* permettaient d'améliorer significativement les résultats des modèles sur la couche de surface atmosphérique par rapport aux méthodes *one-way*. Dans un souci d'obtenir la meilleure solution au problème de couplage, nous nous intéresserons dans la suite de ce document uniquement aux méthodes *two-way*.

Les méthodes de couplage sont différentes stratégies d'implémentation numérique des modèles couplés. Nous les présenterons donc sous un formalisme discrétisé. Notons que la discrétisation des équations (2.9) apporte également son lot de problèmes. En effet, pour des soucis de stabilité, des schémas temporels implicites sont utilisés et peuvent poser problème pour obtenir la stricte égalité des flux sur la fenêtre temporelle. Nous commencerons par un petit point de vocabulaire concernant les différents degrés de couplage puis nous présenterons deux stratégies de couplage usuelles : les méthodes dites *synchrones* et celles dites *asynchrones*. Enfin nous introduirons une méthode de résolution itérative dite de Schwarz (Lemarié *et al.*, 2015).

2.3.1 Vocabulaire du couplage

Le problème du couplage (2.9) ré-écrit dans un formalisme discrétisé est le suivant :

$$\begin{aligned} & \text{Trouver } \mathbf{x}_a \text{ et } \mathbf{x}_o \text{ sur la fenêtre d'intégration } [0, T] \text{ tels que} \\ & \left\{ \begin{array}{ll} M_a(\mathbf{x}_a) = f_a & \text{sur } \Omega_a \times [0, T] \\ M_o(\mathbf{x}_o) = f_o & \text{sur } \Omega_o \times [0, T] \\ D_a(\mathbf{x}_{a|\Gamma}) = D_o(\mathbf{x}_{o|\Gamma}) = F_{oa}(\mathbf{x}_{a|\Gamma}, \mathbf{x}_{o|\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \end{array} \right. \quad (2.10) \end{aligned}$$

Nous définissons alors trois niveaux de couplage correspondant aux solutions $(\mathbf{x}_a, \mathbf{x}_o)$ du problème :

- **librement couplé**¹ → l'égalité $D_a(\mathbf{x}_{a|\Gamma}) = D_o(\mathbf{x}_{o|\Gamma})$ n'est pas respectée sur la fenêtre $[0, T]$, à cause par exemple d'un décalage temporel
- **faiblement couplé** → on vérifie une égalité moyenne des flux sur la fenêtre temporelle : par exemple $\overline{D_a(\mathbf{x}_{a|\Gamma})}^{[0, T]} = D_o(\mathbf{x}_{o|\Gamma})$ où $\overline{\cdot}^{[0, T]}$ est l'opérateur de moyenne temporelle sur la fenêtre $[0, T]$

1. Traduction de l'anglais *loosely coupled*

— **fortement couplé** → nous avons l'égalité stricte des flux sur la fenêtre temporelle :
 $D_a(\mathbf{x}_{a|\Gamma}) = D_o(\mathbf{x}_{o|\Gamma})$ sur $[0, T]$

Il est possible de se référer à l'article de Keyes *et al.* (2013) pour plus de détails concernant les différents niveaux de couplage dans les problèmes multi-physiques. Nous allons à présent présenter différentes stratégies de couplage que nous relierons à ce vocabulaire.

2.3.2 Couplage synchrone au pas de temps

L'objectif des **méthodes synchrones** est d'échanger en permanence des informations entre les deux milieux afin de se rapprocher de la réalité où l'atmosphère et l'océan communiquent en permanence. Cependant, les deux modèles ayant des résolutions temporelles différentes, il est seulement possible de coupler l'océan et l'atmosphère à la fréquence du plus grand pas de temps des deux modèles. Notons Δt_o le pas de temps du modèle océanique et Δt_a celui de l'atmosphère. Considérons que $\Delta t_o = n_{oa}\Delta t_a$ avec $n_{oa} \in \mathbb{N}$, alors la fenêtre d'intégration $[0, T]$ se découpe en N_j fenêtres temporelles $T_j = [t_j, t_j + n_{oa}\Delta t_a] = [t_j, t_j + \Delta t_o] = [t_j, t_{j+1}]$ telles que $[0, T] = \bigcup_{j=0}^{N_j-1} T_j$. L'algorithme de couplage synchrone s'écrit :

$$\begin{cases} M_{a,j,j+1}(\mathbf{x}_{a,j}) = f_a & \text{sur } \Omega_a \times [t_j, t_{j+1}] \\ D_a(\mathbf{x}_{a|\Gamma,i}) = F_{oa}(\mathbf{x}_{a|\Gamma,i}, \mathbf{x}_{o|\Gamma,j}, \mathcal{R}_i) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases}$$

$$\begin{cases} M_{o,j,j+1}(\mathbf{x}_{o,j}) = f_o & \text{sur } \Omega_o \times [t_j, t_{j+1}] \\ D_o(\mathbf{x}_{o|\Gamma,j}) = F_{oa}(\mathbf{x}_{a|\Gamma,j}, \mathbf{x}_{o|\Gamma,j}, \mathcal{R}_j) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases}$$

où les indices \cdot_j correspondent aux indices des pas de temps $t_i \in [t_j, t_{j+1}]$. Les modèles sont intégrés sur la suite de fenêtres temporelles $[t_j, t_{j+1}]$ et échangent des flux instantanés à la fin de chaque fenêtre temporelle. Notons que les schémas d'intégration temporels implicites ajoutent également un décalage temporel dans le calcul et l'échange des flux. C'est pourquoi cette méthode sera qualifiée de **librement couplée**. Un des principaux désavantages de cette technique est le coût de calcul nécessaire pour coupler les deux modèles. En effet, l'échange des flux à chaque pas de temps Δt_o impose de nombreuses communications entre les modèles. Or ces communications peuvent être coûteuses numériquement. Une autre contrainte imposée par ce couplage est la consistance temporelle des deux modèles afin de permettre leur communication tous les n_{oa} pas de temps. Nous pouvons cependant noter que cet algorithme fournit une solution au problème de couplage (2.10) asymptotiquement quand $\Delta t_o \rightarrow 0$ et $\Delta t_a \rightarrow 0$.

2.3.3 Couplage asynchrone par fenêtres temporelles

L'algorithme de **couplage asynchrone** propose une méthode de couplage par fenêtres temporelles, réduisant alors le nombre d'échanges entre les modèles par rapport au couplage synchrone. La fenêtre d'intégration $[0, T]$ est séparée en N_j fenêtres temporelles $T_j = [t_j, t_{j+1}]$ telles que $[0, T] = \bigcup_{j=0}^{N_j-1} [t_j, t_{j+1}]$. Les modèles échangent sur ces fenêtres des flux calculés à

partir des moyennes temporelles des champs :

$$\begin{cases} M_{a,j,j+1}(\mathbf{x}_{a,j}) = f_a & \text{sur } \Omega_a \times [t_j, t_{j+1}] \\ D_a(\mathbf{x}_{a\Gamma,i}) = F_{oa}(\mathbf{x}_{a\Gamma,i}, \overline{x_{o\Gamma}}^{j-1}, \mathcal{R}_i) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases}$$

puis

$$\begin{cases} M_{o,j,j+1}(\mathbf{x}_{o,j}) = f_o & \text{sur } \Omega_o \times [t_j, t_{j+1}] \\ D_o(\mathbf{x}_{o\Gamma,j}) = \overline{D_a(\mathbf{x}_{a\Gamma,i})}^j & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases} \quad (2.11)$$

où $\overline{\cdot}^j$ est l'opérateur de moyenne temporelle sur la fenêtre T_j . Le modèle atmosphérique est alors *en avance* par rapport à l'océan sur la fenêtre temporelle T_j . En effet, le flux F_{oa} est calculé à partir du champ océanique $\overline{x_{o\Gamma}}^{j-1}$ moyenné sur la fenêtre temporelle précédente, puis ce flux est ensuite moyenné sur la fenêtre temporelle pour forcer l'océan. La solution fournie par cet algorithme (2.11) n'est pas la solution exacte du problème de couplage (2.10) du fait de l'asynchronicité de la méthode, mais également à cause de l'utilisation de champs moyennés. Nous parlerons alors également d'une solution **librement couplée**. Bien que ne résolvant pas le problème de couplage (2.10), cette méthode a cependant l'avantage d'avoir un coût numérique relativement faible, nécessitant peu d'échanges entre les modèles. Grâce à ce faible coût numérique, cette méthode de couplage est généralement préférée dans les centres de prévision opérationnels.



Méthodes clefs

Couplage synchrone

Échange des flux océan-atmosphère à chaque pas de temps commun aux deux modèles

- + Résolution du problème de couplage si $\Delta t_o \rightarrow 0$ et $\Delta t_a \rightarrow 0$
- Coût de calcul trop élevé et incompatibilité avec les schémas d'intégration implicite

Couplage asynchrone

Échange des flux océan-atmosphère par fenêtre temporelle

- + Coût de calcul faible
- Décalage temporel du flux

2.3.4 Couplage par méthode itérative de Schwarz

Devant l'incapacité des méthodes synchrones et asynchrones à résoudre le problème de couplage, nous allons nous pencher sur une dernière méthode basée sur les méthodes de décomposition de domaine de Schwarz, initialement connues pour être utilisées dans les problèmes de couplage stationnaires avec différentes physiques ou modélisations numériques. Son utilisation a récemment été étendue aux problèmes temporels avec le développement de la méthode de Schwarz Globale en Temps (voir Gander et Halpern (2007)), aussi appelée méthode *Schwarz Waveform Relaxation* (SWR). L'idée de cette méthode est de résoudre séparément les problèmes sur chacun des domaines (Ω_o et Ω_a) et d'itérer ensuite jusqu'à convergence, c'est à dire jusqu'à obtenir une solution (x_a, x_o) satisfaisant le problème (2.10). Notons qu'à chaque itération a lieu un échange de conditions aux limites entre les modèles. L'avantage de cette méthode est qu'elle permet de trouver une solution couplée au sens fort mathématiquement du problème (2.10). Cette méthode a aussi l'avantage non négligeable de rester non intrusive vis à vis des modèles. Cependant, il s'agit d'une méthode itérative qui peut donc s'avérer très

chère numériquement si sa vitesse de convergence est trop lente. Des études de la vitesse de convergence ont pu être menées, notamment par Lemarié *et al.* (2013a,b) où la convergence de l'algorithme a été évalué en fonction du choix des conditions d'interface entre les modèles. Dans le cadre de notre problème de couplage (2.10), l'algorithme de Schwarz est le suivant :

Pour chaque fenêtre $[t_j, t_{j+1}]$, on itère jusqu'à convergence :

$$\begin{cases} M_{a,j,j+1}(\mathbf{x}_{a,j}^k) = f_a & \text{sur } \Omega_a \times [t_j, t_{j+1}] \\ D_a(\mathbf{x}_{a\Gamma,i}^k) = F_{oa}(\mathbf{x}_{a\Gamma,i}^k, \mathbf{x}_{o\Gamma,i}^{k-1}, \mathcal{R}_i) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases} \quad (2.12)$$

$$\begin{cases} M_{o,j,j+1}(\mathbf{x}_{o,j}^k) = f_o & \text{sur } \Omega_o \times [t_j, t_{j+1}] \\ D_o(\mathbf{x}_{o\Gamma,i}^k) = D_a(\mathbf{x}_{a\Gamma,i}^k) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases}$$

où k désigne l'itération de Schwarz. Nous remarquerons que nous avons besoin de définir une condition initiale $\mathbf{x}_{o\Gamma}^{k=0}$ à l'interface, qui est généralement donnée par la solution de la fenêtre temporelle précédente. Il est possible considérer cet algorithme avec des flux moyens en temps :

Pour chaque fenêtre $[t_j, t_{j+1}]$, on itère jusqu'à convergence :

$$\begin{cases} M_{a,j,j+1}(\mathbf{x}_{a,j}^k) = f_a & \text{sur } \Omega_a \times [t_j, t_{j+1}] \\ D_a(\mathbf{x}_{a\Gamma,i}^k) = F_{oa}(\mathbf{x}_{a\Gamma,i}^k, \overline{\mathbf{x}_{o\Gamma,i}^{k-1}}^j, \mathcal{R}_i) & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases} \quad (2.13)$$

$$\begin{cases} M_{o,j,j+1}(\mathbf{x}_{o,j}^k) = f_o & \text{sur } \Omega_o \times [t_j, t_{j+1}] \\ D_o(\mathbf{x}_{o\Gamma,i}^k) = \overline{D_a(\mathbf{x}_{a\Gamma,i}^k)}^j & \text{sur } \Gamma \times [t_j, t_{j+1}] \end{cases}$$

Dans les deux cas (2.12) et (2.13), l'algorithme de Schwarz est écrit sous sa forme dite *multiplicative*, c'est à dire que chaque modèle est intégré successivement, l'un après l'autre. Si nous remplaçons dans (2.12) par exemple la condition $D_o(\mathbf{x}_{o\Gamma,i}^k) = D_a(\mathbf{x}_{a\Gamma,i}^k)$ par $D_o(\mathbf{x}_{o\Gamma,i}^k) = D_a(\mathbf{x}_{a\Gamma,i}^{k-1})$, nous avons la forme dite *parallèle* de l'algorithme puisque les deux modèles peuvent alors être intégrés simultanément à partir de l'itération précédente. L'algorithme utilisant les flux parfaits (2.12) fournira à convergence une solution fortement couplée au problème (2.10) alors que celui utilisant des flux moyens (2.13) produira une solution faiblement couplée, mais présentera l'avantage d'être moins coûteux numériquement. La figure 2.3 présente un récapitulatif schématique des trois différentes méthodes de couplage introduites.



Méthode clef

Couplage algorithme de Schwarz

Méthode itérative d'échange des flux par fenêtre temporelle

+ Propose une solution forte au problème du couplage

- Coût de calcul potentiellement élevé dû aux itérations

Maintenant que le principe de l'assimilation de données et que le problème du couplage au sein des modèles de prévision ont été présentés dans ces deux premiers chapitres, nous allons pouvoir commencer à nous intéresser à l'assimilation de données couplée. Il s'agit de méthodes d'assimilation de données prenant en compte le problème du couplage afin de produire une analyse consistante à l'interface air-mer. Le prochain chapitre sera dédié à la présentation des enjeux du développement de telles méthodes, et aux solutions proposées depuis les années 90.

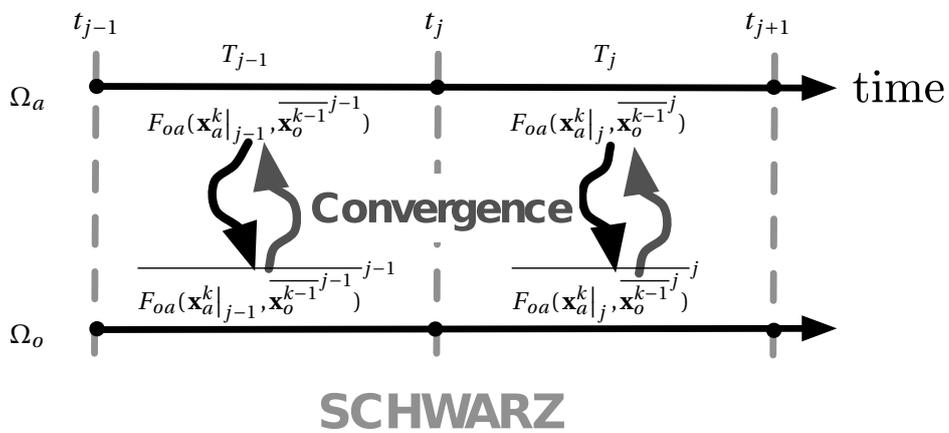
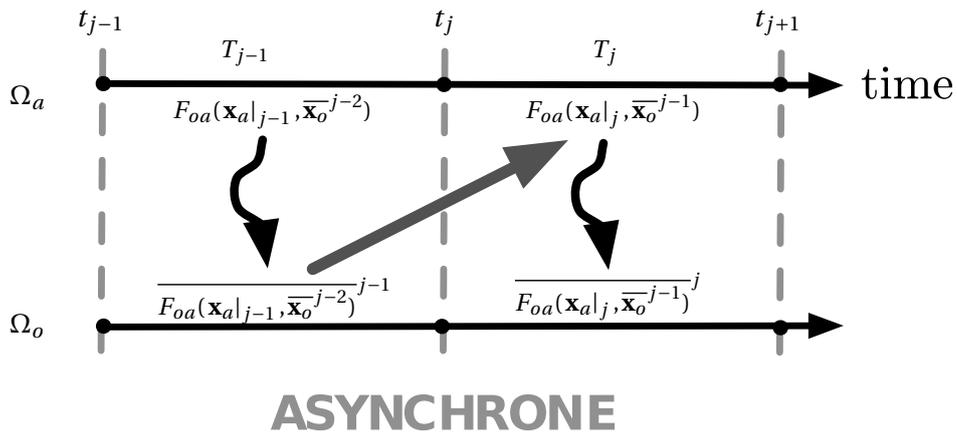
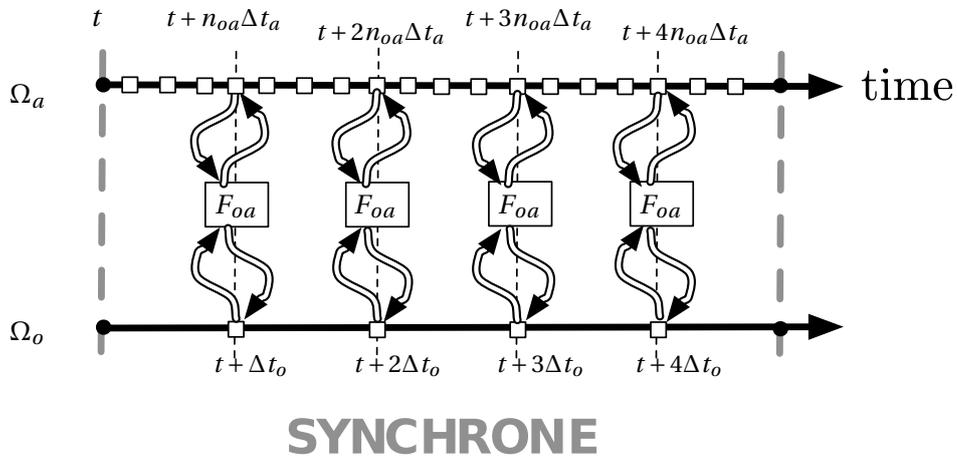


FIGURE 2.3 – Récapitulatif des différentes méthodes de couplage.

Chapitre 3

Histoire du développement de l'assimilation de données couplée

Sommaire

3.1 Motivations et vocabulaire de l'assimilation de données couplée	44
3.1.1 Motivations	44
3.1.2 Vocabulaire de l'assimilation de données couplée	45
3.2 Méthodes d'assimilation de données introduites au fil du temps	46
3.2.1 L'assimilation de données couplée pour la prévision d'ENSO	46
3.2.2 Des méthodes s'adaptant à des dynamiques de différentes échelles . .	48
3.2.3 Recherche d'un algorithme performant au moindre coût	50
3.3 Objectifs	54

Le principe d'assimilation de données couplée a été introduit pour répondre au problème de l'assimilation de données dans le cadre des modèles couplés. En effet, les différents algorithmes d'assimilation introduits dans le chapitre 1 ont été initialement développés pour des modèles découplés, d'atmosphère ou d'océan par exemple. C'est pourquoi depuis les années 1990 des centres de recherche tentent d'adapter les différentes méthodes d'assimilation au problème du couplage, définissant ainsi l'assimilation de données couplée :

Définition 3.0.1 - Assimilation de données couplée

On appelle assimilation de données couplée un algorithme d'assimilation permettant de fournir à un système composé de plusieurs modèles couplés une analyse prenant en compte partiellement ou totalement leur couplage.

Nous exposerons dans ce chapitre les motivations qui poussent à développer des méthodes de plus en plus performantes d'assimilation de données adaptées aux modèles couplés ainsi que le vocabulaire relatif à ce domaine de recherche. Puis nous présenterons un historique des méthodes introduites depuis les années 90 avant d'exposer les objectifs de ces travaux de thèse.

3.1 Motivations et vocabulaire de l'assimilation de données couplée

3.1.1 Motivations

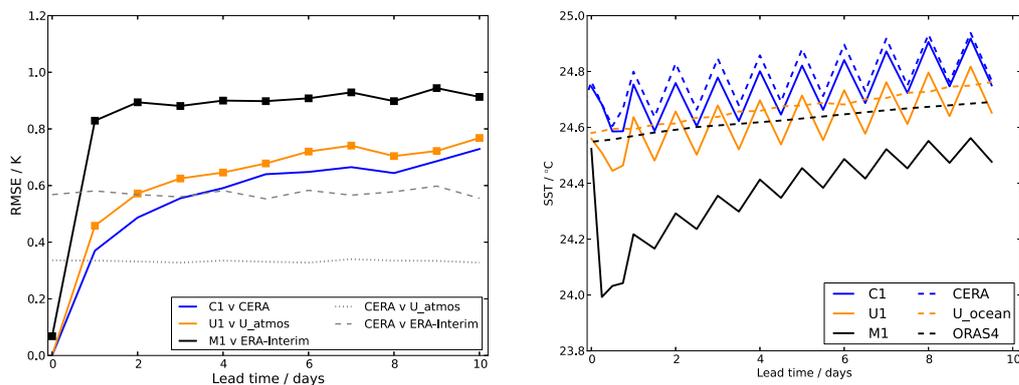
Si un modèle couplé, océan-atmosphère par exemple, est initialisé via des méthodes d'assimilation ne prenant pas en compte le couplage à l'interface air-mer, alors l'état initial ne sera pas en équilibre dynamique et présentera des inconsistances de flux à l'interface. Mulholland *et al.* (2015) ont souhaité évaluer l'impact d'une analyse inconsistante fournie à un modèle couplé. Pour cela ils ont initialisé un modèle de prévision couplé à partir de trois jeux d'analyse différents :

1. **CERA** : une analyse couplée océan ↔ atmosphère produite par le système CERA (Laloyaux *et al.*, 2015).
2. **U_atmos** et **U_ocean** : deux analyses d'atmosphère et d'océan produites par une méthode d'assimilation avec un couplage atmosphère → océan.
3. **ERA-Interim** et **ORAS4** : deux analyses d'atmosphère et d'océan découplées, provenant de deux jeux de réanalyses différents.

Trois prévisions de 10 jours sont ensuite produites avec le modèle couplé à partir des trois jeux d'analyses cités ci-dessus :

1. **C1** une prévision dont l'état initial est fourni par l'analyse CERA
2. **U1** une prévision dont l'état initial est fourni par U_atmos et U_ocean
3. **M1** une prévision dont l'état initial est fourni par ERA-Interim et ORAS4

Les résultats mettent en avant la qualité des prévisions pour C1 par rapport à U1 et M1 sur une région observée. Cela est quantifié par exemple par l'erreur quadratique moyenne (RMSE) des prévisions par rapport à leur analyse de référence en figure 3.1a. On y observe que la prévision C1 présente une RMSE inférieure à U1 qui elle-même est inférieure à M1 sur toute la fenêtre temporelle de la prévision. De plus, on peut observer dans la figure 3.1b la présence d'un fort choc dans les premiers pas de temps de l'évolution de la température de surface de l'océan au début de la fenêtre temporelle pour la prévision M1.



(a) RMSE moyennée sur la région observée de la température de surface de l'atmosphère. (b) Température de surface de l'océan moyenne sur la région observée.

FIGURE 3.1 – La figure (a) présente la RMSE des prévisions par rapport à leurs analyses respectives, ainsi que la RMSE entre les analyses. La figure (b) met en avant le choc à l'initialisation de la prévision M1. D'après Mulholland *et al.* (2015).

Cette étude démontre la meilleure qualité des prévisions disposant d'un état initial en équilibre dynamique à l'interface et que des chocs d'initialisation peuvent survenir lorsqu'une analyse inconsistante est fournie à un modèle couplé. Ces chocs sont caractérisés par de forts ajustements du modèle lors des premiers pas de temps créant alors des erreurs de prévision. Ils peuvent être provoqués par le déséquilibre des flux verticaux à l'interface des modèles ou bien par l'utilisation d'un modèle de prévision différent de celui utilisé pour créer l'analyse. Il existe d'autres études démontrant l'intérêt des méthodes d'assimilation de données couplée vis à vis de leurs homologues non-couplée. Nous pouvons citer par exemple Zhang (2011) qui évalue de manière positive l'impact des méthodes d'assimilation de données couplée pour la calibration de paramètres de modèles, ou encore Chen (2012) qui met en avant leur influence positive pour la prévision de l'Oscillation Australe El-Niño (ENSO). Tardif *et al.* (2013) ou Liu *et al.* (2013) ont également montré que les méthodes d'assimilation couplée permettaient la propagation des covariances d'erreur d'un modèle à l'autre. Cela permet notamment d'avoir des informations sur l'état de l'océan en l'absence d'observations dans ce milieu.



Motivations à l'assimilation de données couplée

Dans le cadre des modèles couplés :

- Réduction des chocs à l'initialisation
- Amélioration des prévisions
- Influence d'une observation d'un milieu à l'autre grâce aux covariances croisées
- Calibration de paramètres plus performante

3.1.2 Vocabulaire de l'assimilation de données couplée

Avant de présenter diverses méthodes d'assimilation de données couplée, il nous semble important de définir quelques notions permettant de juger de la *force* du couplage dans les algorithmes, qui peut intervenir de deux manières différentes :

- au sein du modèle direct utilisé,
- au cœur de l'assimilation de données grâce à des covariances croisées et/ou l'adjoint d'un modèle couplé.

Des notions d'assimilation fortement et faiblement couplées ont alors été introduites dans divers papiers, notamment dans celui de Smith *et al.* (2015). *Fortement couplé* fait référence à un algorithme d'assimilation où le couplage intervient des deux manières citées ci-dessus, c'est à dire au sein du modèle direct utilisé et au cœur de l'assimilation de données. En revanche, un algorithme *faiblement couplé* prendra uniquement en compte le couplage dans le modèle direct. Ce vocabulaire soulève cependant une ambiguïté avec la qualité du couplage de l'analyse. En effet, chacun remarquera que la définition ci-dessus d'un algorithme fortement couplé ne fait pas référence au couplage présent dans le modèle. De plus, le couplage peut être pris en compte de nombreuses manières dans un processus d'assimilation (Laloyaux, 2016) :

- par l'impact d'une observation d'un milieu à l'autre
- par des covariances d'erreurs croisées d'un modèle à l'autre
- par l'utilisation de modèles tangent et adjoint couplés...

Les seuls adjectifs de *faiblement* et *fortement* couplés ne permettent donc pas de se rendre compte du couplage présent au sein d'un algorithme.

⚠ Assimilation de données fortement couplée n'est pas synonyme de la création d'une analyse fortement couplée au sens mathématique du terme.

Nous prendrons donc soin dans la suite de préciser le niveau de couplage présent au sein de l'assimilation et au sein des modèles. Nous pourrons alors évaluer le couplage d'une méthode d'assimilation sous deux angles, du point de vue de l'algorithme d'assimilation et du modèle utilisé :

Algorithme d'assimilation	Modèle
Découplé	Deux modèles découplés
↓	↓
Faiblement couplé	Librement couplés
↓	↓
Fortement couplé	Fortement couplés

Tableau 3.1 – Différents degrés de couplage dans un algorithme d'assimilation de données, du point de vue algorithme et du point de vue modèle

Remarque 3.1.1

Nous avons omis ici de mentionner les modèles faiblement couplés puisque nous ne serons pas amenés à en rencontrer dans la suite de ce manuscrit.

3.2 Méthodes d'assimilation de données introduites au fil du temps

3.2.1 L'assimilation de données couplée pour la prévision d'ENSO

Un premier enjeu pour l'assimilation couplée qui a été cerné est la prévision d'ENSO. En effet, il a pu être mis en évidence que l'initialisation des modèles jouait un rôle majeur dans la qualité des prévisions de ce phénomène (Cane *et al.*, 1986; Latif *et al.*, 1994). Chen *et al.* (1995, 1997) sont à notre connaissance les premiers à s'être intéressés à l'assimilation de données couplée pour initialiser un modèle couplé. Il s'agit d'un modèle simplifié d'anomalies, couplé de manière synchrone, dont la dynamique est gouvernée par des équations *shallow-water* linéaires forcées par différents traceurs de manière non-linéaire (modèle CZ (Zebiak et Cane, 1987)). Ils utilisent ce modèle couplé au sein d'un algorithme de nudging (1.2) :

Assimilation de données couplée par nudging (Chen *et al.*, 1995, 1997)

$$\begin{aligned} \text{Initialisation} \quad & \mathbf{x}_0^b = (\mathbf{x}_{a,0}^b, \mathbf{x}_{o,0}^b)^T \\ \text{Étape d'analyse} \quad & \mathbf{x}_{a,i}^a = \mathbf{x}_{a,i}^b + \mathbf{K}(\mathbf{y}_i^o - \mathbf{H}\mathbf{x}_{a,i}^b) \\ \text{Étape de prévision} \quad & \mathbf{x}_{i+1}^b = M_{i,i+1}\mathbf{x}_i^a \end{aligned}$$

- \mathbf{y}_i^o sont des observations des anomalies de tension de vent.
- Le gain \mathbf{K} est un coefficient variable en fonction de la latitude.

L'analyse créée par cet algorithme est alors plus consistante à l'interface et présente une variabilité temporelle et spatiale plus proche de celle du modèle qu'une analyse classique. La prédictibilité du phénomène ENSO a alors été améliorée. Nous citerons également Rosati *et al.* (1997) qui montrent que l'ajout d'informations provenant de l'atmosphère pour l'assimilation des données de l'océan permet d'améliorer les prévisions d'ENSO. Ces trois premières études (Chen *et al.*, 1995, 1997; Rosati *et al.*, 1997) sont les premières à notre connaissance à s'intéresser à la création d'une analyse plus consistante à l'interface océan-atmosphère pour des modèles couplés. Elles présentent tout de même quelques limitations :

- Chen *et al.* (1995) se limitent au nudging des seules anomalies de tension de vent
- Rosati *et al.* (1997) utilisent des méthodes, observations et modèles différents pour fonder leurs résultats → incertitudes quant aux facteurs expliquant une analyse de bonne ou mauvaise qualité.

Nous pouvons ensuite citer Lee *et al.* (2000) qui tentent d'assimiler des données de manière couplée avec un 4D-Var. La fonction coût mesure les écarts entre le modèle (couplé de manière synchrone) et des observations d'anomalies de température et de hauteur de surface côté océan, et de tension du vent de surface côté atmosphère. Le vecteur de contrôle du 4D-Var est composé de l'état initial et de paramètres Φ du modèle couplé. La minimisation est réalisée grâce à la méthode adjointe :

Assimilation de données couplée par un 4D-Var (Lee *et al.*, 2000)

Algorithme du 4D-Var (1.3) avec :

- Vecteur d'état $\mathbf{x}_0 = (\mathbf{x}_{a,0}, \mathbf{x}_{o,0}, \Phi)^T$
- Modèle direct M couplé
- Fonction coût :

$$J = \sum_{i=0}^N (\mathbf{y}_{o,i}^o - H_{o,i} M_{0,i} \mathbf{x}_0)^T \mathbf{R}_{o,i}^{-1} (\mathbf{y}_{o,i}^o - H_{o,i} M_{0,i} \mathbf{x}_0) + \sum_{i=0}^N (\mathbf{y}_{a,i}^o - H_{a,i} M_{0,i} \mathbf{x}_0)^T \mathbf{R}_{a,i}^{-1} (\mathbf{y}_{a,i}^o - H_{a,i} M_{0,i} \mathbf{x}_0)$$

- Évaluation de ∇J avec les modèles linéaires et adjoints couplés : \mathbf{M} et \mathbf{M}^T

L'utilisation du modèle adjoint couplé au cours de la minimisation permet de prendre en compte la dynamique du couplage océan-atmosphère pour l'estimation de l'analyse. Leurs résultats montrent une amélioration des prévisions d'ENSO et une réduction des chocs à l'initialisation. Galanti *et al.* (2003) présentent une méthode très similaire avec un algorithme type 4D-Var utilisant un modèle couplé très simplifié. L'adjoint du modèle couplé est utilisé pour minimiser des écarts de température de sous-couche et de surface océanique ainsi que de tension de vent. Comme Lee *et al.* (2000), ils mettent en avant l'amélioration induite par la prise en compte de la dynamique du couplage océan-atmosphère par le 4D-Var grâce à l'adjoint du modèle couplé. Bien que ces méthodes assimilent un plus grand nombre de données que Chen *et al.* (1995), nous pouvons cependant regretter l'absence dans ces deux méthodes d'un terme de rappel vers l'ébauche J^b . Cela suppose que les seules observations suffisent à estimer un état initial cohérent, or cela n'est pas réaliste en pratique comme nous avons pu le mentionner dans le chapitre 1.

Enfin, Sun *et al.* (2002) présentent une méthode utilisant un filtre de Kalman étendu (EKF) avec un modèle non-linéaire simplifié d'océan et d'atmosphère couplé de manière synchrone. Ils comparent deux méthodes d'assimilation :

- une méthode utilisant un modèle découplé au sein de l'EKF (filtre découplé)
- une méthode utilisant un modèle couplé au sein de l'EKF (filtre couplé)

Il ressort de cette étude que le filtre couplé propage de manière plus significative les covariances d'erreurs entre les modèles, notamment dans l'océan. Il permet donc une plus forte corrélation entre les différents milieux lors de l'assimilation que le filtre découplé, améliorant a posteriori les prévisions du modèle. Kondrashov *et al.* (2008) ont étendu cette étude avec l'estimation de paramètres prise en compte dans le filtre couplé. Ils montrent que l'estimation de paramètres jointe à l'estimation de l'état initial joue un rôle important dans la qualité des prévisions du modèle couplé. Notons cependant qu'un tel algorithme est inutilisable en pratique à cause des limitations numériques de l'EKF (voir chapitre 1).

Zhang *et al.* (2007) proposent alors un système d'assimilation de données couplée utilisant un filtre de Kalman d'ensemble : le Filtre de Kalman d'Ensemble Ajusté (EAKF) (Anderson, 2001), utilisable dans un cadre réaliste. Ils utilisent un modèle couplé de manière synchrone pour estimer les covariances d'erreur d'ébauche entre les différentes variables du modèle. Des covariances d'erreurs croisées entre l'océan et l'atmosphère sont ainsi créées grâce au couplage interne du modèle. Les observations atmosphériques semblent avoir un rôle majeur dans la qualité de l'analyse, mettant en avant un impact positif de l'assimilation couplée, spécifiquement dans la région équatoriale-tropicale pour la prévision d'ENSO.



L'assimilation couplée pour ENSO

Premières applications de l'assimilation de donnée couplée : la prévision d'ENSO

Chen *et al.* (1995, 1997) commencent par proposer une méthode de nudging utilisant un modèle couplé.

Cadre variationnel :

- Lee *et al.* (2000) et Galanti *et al.* (2003) proposent un 4D-Var sans terme d'ébauche avec un modèle direct, tangent et adjoint couplé.

Cadre stochastique :

- Sun *et al.* (2002) et Kondrashov *et al.* (2008) utilisent un modèle couplé au sein d'un EKF.
- Zhang *et al.* (2007) proposent une méthode d'assimilation couplée au sein d'une variante de l'EnKF.

3.2.2 Des méthodes s'adaptant à des dynamiques de différentes échelles

Bien que l'adaptation des méthodes usuelles d'assimilation au cadre couplé ait pu montrer des résultats encourageants, Sugiura *et al.* (2008), Lu *et al.* (2015) et Ballabrera-Poy *et al.* (2009) constatent qu'assimiler simultanément des données haute et basse fréquence, telles qu'atmosphériques et océaniques, peut poser problème. En effet, ils observent que des corrélations croisées non-désirées peuvent se créer entre les milieux, dégradant alors les résultats. Deux stratégies sont alors envisagées et présentées ici.

Un filtrage haute fréquence des données

Pour étudier ce problème, Sugiura *et al.* (2008) choisissent d'utiliser un algorithme de type 4D-Var avec un modèle couplé de manière asynchrone. Ils proposent d'utiliser des champs moyennés sur 10 jours, faisant ressortir le couplage basse fréquence entre l'océan et l'atmosphère. Leurs résultats montrent de nettes améliorations pour les prévisions d'ENSO. Cela validerait leur choix d'utiliser des données moyennées dans le cadre de prévisions saisonnières faisant intervenir des dynamiques aux échelles différentes. Lu *et al.* (2015) proposent une méthode équivalente au sein d'un filtre EnKF, en lissant les observations et l'ébauche de l'atmosphère. Les corrélations croisées entre les milieux sont alors améliorées. Bien que ces deux méthodes semblent proposer un moyen de résoudre le problème de différence d'échelles entre les milieux, l'utilisation de données moyennées implique la perte de toute la dynamique petite échelle présente à l'interface air-mer. Le problème, plutôt que d'être résolu, est en fait contourné en simplifiant drastiquement le couplage océan-atmosphère.

Une méthode hybride plus adaptée?

Plutôt que de moyenner les champs, Ballabrera-Poy *et al.* (2009) cherchent à développer un système d'assimilation de données flexible, combinant faible et haute résolution. Leur modèle est constitué de deux équations de Lorenz (Lorenz et Emanuel, 1998) représentant 256 variables *rapides* (atmosphériques) et 8 *lentes* (océaniques). Ils choisissent d'initialiser les variables océaniques via un filtre EnKF prenant en compte le système couplé, et les variables atmosphériques sont contraintes par une méthode de nudging.

Assimilation de données couplée hybride (Ballabrera-Poy *et al.*, 2009)

Algorithme d'ensemble (m dénote les membres de l'ensemble)

Initialisation	$\mathbf{x}_{0,m}^b = (\mathbf{x}_{a,0,m}^b, \mathbf{x}_{o,0,m}^b)^T = (\mathbf{x}_{a,0}^b, \mathbf{x}_{o,0}^b)^T + \eta_{0,m}$
Étape d'analyse	$\mathbf{x}_{o,i,m}^a$ Analyse de type EnKF (1.4) (Océan) ↳ Covariances et gain de Kalman évalués à partir du système couplé
	$\mathbf{x}_{a,i,m}^a$ Analyse par nudging (1.2) (Atmosphère)
Étape de prévision	$\mathbf{x}_{i+1,m}^b = M_{i,i+1} \mathbf{x}_{i,m}^a + \eta_{i,m}$ avec M le modèle couplé

La définition des ensembles, notamment du bruit $\eta_{i,k}$, est décrite dans Ballabrera-Poy *et al.* (2009). Les résultats de cette méthode hybride sont meilleurs que ceux où seul un filtre EnKF était utilisé pour initialiser l'ensemble des variables. En effet, l'EnKF créait des covariances croisées indésirables lorsque pas ou trop peu d'observations atmosphériques étaient présentes. L'utilisation d'une méthode de nudging pour le modèle atmosphérique est cependant une forte limitation de cette méthode car cela pourrait produire une analyse non-consistante avec la dynamique du modèle. De plus, cette méthode a été testée uniquement sur un modèle académique.



Des limitations à l'assimilation couplée?

Les dynamiques très différentes de l'océan et l'atmosphère peuvent poser problème

Deux pistes sont envisagées :

- Un filtrage haute fréquence des données (Sugiura *et al.*, 2008; Lu *et al.*, 2015)
-  Perte du signal haute fréquence de l'atmosphère
- Une méthode hybride d'assimilation pour mieux prendre en compte les signaux haute et basse fréquence (Ballabrera-Poy *et al.*, 2009)

3.2.3 Recherche d'un algorithme performant au moindre coût

Les différentes solutions proposées au problème de l'assimilation de données couplée n'étant pas satisfaisantes, de nombreux efforts sont fournis ces dernières années pour développer de nouvelles méthodes. En effet, il y a donc un besoin des centres météorologiques opérationnels à disposer de méthodes d'assimilation couplée performantes, tout en conservant un coût de calcul raisonnable. Nous présentons ici les dernières méthodes proposées, stochastiques et variationnelles.

Utilisation des méthodes stochastiques

Luo et Hoteit (2014) et Frolov *et al.* (2016) présentent deux méthodes similaires d'assimilation de données couplée. Les premiers l'appliquent à un modèle de Lorenz 96 (Lorenz et Emanuel, 1998) à 80 variables tandis que Frolov *et al.* (2016) utilisent un modèle mésoéchelle plus réaliste. Dans les deux cas, la propagation de l'état du système est réalisée via le modèle couplé et l'étape d'analyse est effectuée pour l'atmosphère et l'océan séparément. Le couplage au sein de l'analyse est assuré via la présence de covariances croisées. Pour décrire leurs algorithmes, la décomposition matricielle suivante est utilisée :

$$\begin{aligned} \mathbf{x} &= (\mathbf{x}_a, \mathbf{x}_o)^T & \mathbf{y}^o &= (\mathbf{y}_a^o, \mathbf{y}_o^o)^T \\ \mathbf{B} &= \begin{pmatrix} \mathbf{B}_{aa} & \mathbf{B}_{ao} \\ \mathbf{B}_{oa} & \mathbf{B}_{oo} \end{pmatrix} & \mathbf{R} &= \begin{pmatrix} \mathbf{R}_{aa} & \mathbf{R}_{ao} \\ \mathbf{R}_{oa} & \mathbf{R}_{oo} \end{pmatrix} & \mathbf{H} &= \begin{pmatrix} \mathbf{H}_{aa} & \mathbf{H}_{ao} \\ \mathbf{H}_{oa} & \mathbf{H}_{oo} \end{pmatrix} \end{aligned} \quad (3.1)$$

où \mathbf{x} et \mathbf{y}^o sont les vecteurs d'état et d'observation du système couplé que l'on décompose en une partie atmosphérique et une océanique (indices a et o respectivement). \mathbf{B} , \mathbf{R} sont les matrices de covariance d'ébauche et d'observation du problème couplé considéré dans son ensemble, écrites sous forme de matrices blocs pour faire apparaître les covariances croisées entre les modèles (\mathbf{B}_{ao} , \mathbf{B}_{oa} , \mathbf{R}_{ao} et \mathbf{R}_{oa}). \mathbf{H} se décompose de manière équivalente.

Luo et Hoteit (2014) considèrent dans leur cas que $\mathbf{B}_{ao,oa} = \mathbf{R}_{ao,oa} = 0$. En revanche, ils décomposent le gain de Kalman de manière similaire à \mathbf{B} et \mathbf{R} :

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{aa} & \mathbf{K}_{ao} \\ \mathbf{K}_{oa} & \mathbf{K}_{oo} \end{pmatrix}$$

L'étape d'analyse de leur algorithme est alors la suivante :

Assimilation de données couplée avec analyse séparée (Luo et Hoteit, 2014)

$$\mathbf{B}_{ao,oa} = \mathbf{R}_{ao,oa} = 0$$

$$\begin{aligned} \text{Étape d'analyse} \quad \mathbf{x}_{a,i}^a &= \mathbf{x}_{a,i}^b + \mathbf{K}_{aa}(\mathbf{y}_{a,i}^o - \mathbf{H}_{a,i}\mathbf{x}_{a,i}^b) + \mathbf{K}_{ao}(\mathbf{y}_{o,i}^o - \mathbf{H}_{o,i}\mathbf{x}_{o,i}^b) \\ \mathbf{x}_{o,i}^a &= \mathbf{x}_{o,i}^b + \mathbf{K}_{oa}(\mathbf{y}_{a,i}^o - \mathbf{H}_{a,i}\mathbf{x}_{a,i}^b) + \mathbf{K}_{oo}(\mathbf{y}_{o,i}^o - \mathbf{H}_{o,i}\mathbf{x}_{o,i}^b) \end{aligned}$$

Le couplage entre les domaines pour chaque analyse est assuré par les gains $\mathbf{K}_{aa,oo,ao,oa}$ qui font intervenir des termes de chacun des deux milieux. Voir l'étude de Luo et Hoteit (2014) pour le calcul explicite de ces termes avec l'utilisation du filtre de Kalman d'ensemble ETKF (Bishop *et al.*, 2001).

La méthode de Frolov *et al.* (2016) suppose que seules les variables proches de l'interface dans un milieu ont un impact sur l'autre milieu. La décomposition de matrice \mathbf{B} introduite dans l'équation (3.1) est alors étendue à :

$$\begin{pmatrix} \mathbf{B}_{aa} & \mathbf{B}_{ao} \\ \mathbf{B}_{oa} & \mathbf{B}_{oo} \end{pmatrix} \approx \begin{pmatrix} \mathbf{B}_{a_{lib}a_{lib}} & \mathbf{B}_{a_{lib}a_{itf}} & 0 & 0 \\ \mathbf{B}_{a_{itf}a_{lib}} & \mathbf{B}_{a_{itf}a_{itf}} & \mathbf{B}_{a_{itf}o_{itf}} & 0 \\ 0 & \mathbf{B}_{o_{itf}a_{itf}} & \mathbf{B}_{o_{itf}o_{itf}} & \mathbf{B}_{o_{itf}o_{pfd}} \\ 0 & 0 & \mathbf{B}_{o_{pfd}o_{itf}} & \mathbf{B}_{o_{pfd}o_{pfd}} \end{pmatrix}$$

où les indices a_{lib} et a_{itf} permettent de considérer les variables de l'atmosphère libre et les variables proches de l'interface respectivement, de même que o_{itf} et o_{pfd} dénotent les variables océaniques proche de l'interface et en océan profond. Les matrices \mathbf{H} et \mathbf{R} sont également décomposées selon ce principe ainsi que les vecteurs d'états et d'observation :

$$\begin{aligned} \mathbf{x}_a &= \begin{pmatrix} \mathbf{x}_{a_{lib}} \\ \mathbf{x}_{a_{itf}} \end{pmatrix} & \mathbf{x}_o &= \begin{pmatrix} \mathbf{x}_{o_{itf}} \\ \mathbf{x}_{o_{pfd}} \end{pmatrix} \\ \mathbf{y} &= \begin{pmatrix} \mathbf{y}_{a_{lib}} \\ \mathbf{y}_{a_{itf}} \\ \mathbf{y}_{o_{itf}} \\ \mathbf{y}_{o_{pfd}} \end{pmatrix} \end{aligned}$$

On se référera à l'article de Frolov *et al.* (2016) pour le détail des matrices blocs. Enfin, on notera par exemple $\mathbf{H}_{a,o_{itf}}$ la matrice \mathbf{H} telle que ses blocs relatifs aux variables o_{pfd} sont nuls :

$$\mathbf{H}_{a,o_{itf}} = \begin{pmatrix} \mathbf{H}_{a_{lib}a_{lib}} & \mathbf{H}_{a_{lib}a_{itf}} & 0 & 0 \\ \mathbf{H}_{a_{itf}a_{lib}} & \mathbf{H}_{a_{itf}a_{itf}} & \mathbf{H}_{a_{itf}o_{itf}} & 0 \\ 0 & \mathbf{H}_{o_{itf}a_{itf}} & \mathbf{H}_{o_{itf}o_{itf}} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

et cette notation est étendue à l'indice a_{itf} , pour la matrice \mathbf{R} et le vecteur d'observation. L'étape d'analyse proposée par Frolov *et al.* (2016) est alors la suivante :

Assimilation de données couplée par un solveur d'interface (Frolov *et al.*, 2016)

$$\begin{aligned} \text{Étape d'analyse} \quad \mathbf{K}_a &= [\mathbf{B}_{aa} \quad \mathbf{B}_{ao}] \mathbf{H}_{a,o_{itf}}^T (\mathbf{H}_{a,o_{itf}} \mathbf{B} \mathbf{H}_{a,o_{itf}}^T + \mathbf{R}_{a,o_{itf}})^{-1} \\ \mathbf{x}_a^a &= \mathbf{x}_a^b + \mathbf{K}_a (\mathbf{y}_{a,o_{itf}}^o - \mathbf{H}_{a,o_{itf}} \mathbf{x}_a^b) \\ \\ \mathbf{K}_o &= [\mathbf{B}_{oa} \quad \mathbf{B}_{oo}] \mathbf{H}_{a_{itf},o}^T (\mathbf{H}_{a_{itf},o} \mathbf{B} \mathbf{H}_{a_{itf},o}^T + \mathbf{R}_{a_{itf},o})^{-1} \\ \mathbf{x}_o^a &= \mathbf{x}_o^b + \mathbf{K}_o (\mathbf{y}_{a_{itf},o}^o - \mathbf{H}_{a_{itf},o} \mathbf{x}_o^b) \end{aligned}$$

En écrivant l'étape d'analyse ainsi, Frolov *et al.* (2016) proposent d'assimiler des données de manière couplée en ne considérant que le couplage entre des variables proches de l'interface. Un couplage entre l'océan profond et l'atmosphère libre est cependant présent de manière indirecte. En effet, nous pouvons voir le couplage entre les milieux de cette manière :

$$\text{Atm}_{\text{lib}} \leftrightarrow \text{Atm}_{\text{itf}} \leftrightarrow \text{Oce}_{\text{itf}} \leftrightarrow \text{Oce}_{\text{pfd}}$$

où Atm_{lib} désigne l'atmosphère libre (Atm_{itf} , Oce_{itf} , Oce_{pfd} désignent les autres milieux de manière explicite). Cette méthode permet également de renforcer le couplage par rapport à un algorithme tel que celui de Luo et Hoteit (2014) grâce à la présence des termes non diagonaux $\mathbf{B}_{ao,oa}$, tout en respectant un coût de calcul raisonnable. Les résultats montrent que l'algorithme converge vers la solution exhaustive (analyse couplée sur tout le domaine) si suffisamment d'observations sont présentes à l'interface. Cet algorithme présente l'avantage d'être assez flexible, les deux étapes d'analyse étant indépendantes, mais présente la difficulté de définir les matrices de covariance croisées proche de l'interface. Notons que Smith *et al.* (2017) proposent une estimation de cette matrice \mathbf{B} à partir d'un ensemble d'ébauches, comme le ferait un EnKF, puis assimilent ensuite les données avec un 4D-Var incrémental. Ils mettent alors en avant les améliorations induites par cette définition de \mathbf{B} , créant de fortes corrélations croisées entre les milieux proches de l'interface, vis à vis d'une matrice \mathbf{B} sans corrélations croisées.

4D-var incrémental adapté aux problèmes couplés

Smith *et al.* (2015) et Laloyaux *et al.* (2015) ont proposé quant à eux des méthodes d'assimilation de données couplée utilisant un 4D-Var incrémental. Laloyaux *et al.* (2015) présentent un algorithme où un modèle opérationnel couplé de manière asynchrone est utilisé en boucle externe, et où la minimisation est réalisée de manière découplée sur les différents sous-systèmes du modèle en boucle interne :

Algorithme CERA Laloyaux *et al.* (2015)

Algorithme 3.1 4D-Var incrémental avec deux boucles externes

INITIALISATION : $\mathbf{x}_0^r = \mathbf{x}_0^b$

BOUCLE EXTERNE

Intégration du modèle non-linéaire : $\mathbf{x}_i^r = M_{0 \rightarrow i}(\mathbf{x}_0^r)$

Calcul des innovations $d_i = \mathbf{y}_i^o - H_i(\mathbf{x}_i^r)$

BOUCLE INTERNE

ATMOSPHÈRE : Calcul de J_a et ∇J_a grâce à \mathbf{M}_a , \mathbf{M}_a^T , \mathbf{H}_a et \mathbf{H}_a^T

Minimisation : évaluation de $\delta \mathbf{x}_{a,0}^a$

OCÉAN : Calcul de J_o et ∇J_o grâce à \mathbf{M}_o , \mathbf{M}_o^T , \mathbf{H}_o et \mathbf{H}_o^T

Minimisation : évaluation de $\delta \mathbf{x}_{o,0}^a$

FIN DE LA BOUCLE INTERNE

Mise à jour de l'état de référence $\mathbf{x}_0^r = \mathbf{x}_0^r + \delta \mathbf{x}_0^a$ avec $\delta \mathbf{x}_0^a = \left(\delta \mathbf{x}_{a,0}^a{}^T, \delta \mathbf{x}_{o,0}^a{}^T \right)^T$

FIN DE LA BOUCLE EXTERNE

Calcul de l'état d'analyse $\mathbf{x}_0^a = \mathbf{x}_0^r$ et $\mathbf{x}_i^a = M_{0 \rightarrow i}(\mathbf{x}_0^a)$

En ajoutant une seconde itération de boucle externe, le système, appelé CERA, permet une propagation de l'information d'un milieu à l'autre avec la création de corrélations croisées. En effet ré-intégrer le modèle couplé après un premier incrément permet de transférer l'information donnée par cet incrément d'un modèle à l'autre grâce à la dynamique du modèle direct couplé. Le système CERA présente donc de nettes améliorations par rapport à une assimilation découplée que cela soit dans l'océan ou l'atmosphère, et ce pour un coût de développement bien moindre qu'un système utilisant l'adjoint du couplage au sein des boucles internes. Cette méthode nécessite cependant d'effectuer deux boucles externes, donc d'intégrer deux fois le modèle couplé, ce qui peut-être relativement coûteux. Smith *et al.* (2015) ont quant à eux souhaité comparer ce qu'ils appellent un 4D-var incrémental fortement couplé, un faiblement couplé et un non-couplé :

Algorithmes 4D-Var incrémentaux Smith *et al.* (2015)

Algorithme 3.2 4D-Var incrémental avec une seule boucle externe

INITIALISATION : $\mathbf{x}_0 = \mathbf{x}_0^b$

Intégration du modèle non-linéaire : $\mathbf{x}_i^b = M_{0,i}(\mathbf{x}_0^b)$

Calcul des innovations $d_i = \mathbf{y}_i^o - H_i \mathbf{x}_i^b$

BOUCLE INTERNE

$$J(\delta \mathbf{x}_0) = \frac{1}{2} \delta \mathbf{x}_0^T \mathbf{B}^{-1} \delta \mathbf{x}_0 + \frac{1}{2} \sum_{i=0}^N (d_i - \mathbf{H}_i \mathbf{M}_{0,i} \delta \mathbf{x}_0)^T \mathbf{R}_i^{-1} (d_i - \mathbf{H}_i \mathbf{M}_{0,i} \delta \mathbf{x}_0)$$

$$\nabla J(\delta \mathbf{x}_0) = \mathbf{B}^{-1} \delta \mathbf{x}_0 - \sum_{i=0}^N \mathbf{M}_1^T \dots \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} (d_i - \mathbf{H}_i \mathbf{M}_{0,i} \delta \mathbf{x}_0)$$

Minimisation : évaluation de l'incrément d'analyse $\delta \mathbf{x}_0^a$

FIN DE LA BOUCLE INTERNE

Calcul de l'état d'analyse $\mathbf{x}_0^a = \mathbf{x}_0^b + \delta \mathbf{x}_0^a$ et $\mathbf{x}_i^a = M_{0 \rightarrow i}(\mathbf{x}_0^a)$

- Fortement couplé : M , \mathbf{M} et \mathbf{M}^T couplés
- Faiblement couplé : M couplé, \mathbf{M} et \mathbf{M}^T non-couplés
Dédoublage de la fonction coût et minimisation pour évaluer $\delta \mathbf{x}_{a,0}^a$ et $\delta \mathbf{x}_{o,0}^a$
- Non couplé : M , \mathbf{M} et \mathbf{M}^T non-couplés
Dédoublage de la fonction coût et minimisation pour évaluer $\delta \mathbf{x}_{a,0}^a$ et $\delta \mathbf{x}_{o,0}^a$

Ils testent alors ces trois algorithmes avec un modèle colonne couplé de manière asynchrone. La méthode fortement couplée permet de créer des covariances croisées entre les deux milieux grâce au modèle adjoint couplé. Il en résulte une réduction des chocs à l'initialisation, présents dans le cas de la méthode non-couplée. La méthode faiblement couplée, similaire au CERA, ne semble en revanche pas très performante et n'améliore que très peu les résultats par rapport au cas non-couplé. Cela n'est cependant pas très étonnant puisqu'ils n'effectuent qu'une seule boucle externe, à contrario du CERA qui en réalise deux. Les covariances d'erreurs ne peuvent alors pas être propagées par l'intégration du modèle couplé.

Bien que ces méthodes semblent prometteuses, Fowler et Lawless (2016) pointent l'impact négatif des erreurs de modèle sur la qualité de l'analyse. Ils montrent que le couplage au sein de l'assimilation de données permet la propagation de ces erreurs entre les deux milieux, dégradant alors l'analyse pour une meilleure consistance à l'interface. La méthode dite du 4D-Var

Faible Contrainte pourrait alors être une solution à ce problème (Vidard *et al.*, 2000; Griffith et Nichols, 2000; Trémolet, 2006), proposant une solution en estimant ces erreurs en plus de la condition initiale. Il est cependant compliqué en pratique de bien les corriger.



Réduire le coût du couplage dans l'assimilation de données couplée

Le couplage au sein de l'assimilation de donnée peut être très coûteux s'il est pris en compte dans sa globalité (covariances croisées sur tout le système, adjoint du couplage...)

Deux solutions sont proposées pour réduire ce coût tout en gardant une *bonne* qualité d'analyse :

- Ne considérer que les covariances croisées proches de l'interface, qui sont supposées dominer la dynamique du couplage (Frolov *et al.*, 2016).
- Supprimer le couplage dans la boucle interne du 4D-Var incrémental. Le couplage est réalisé dans la boucle externe seulement, permettant tout de même la transmission d'informations d'un milieu à l'autre (Smith *et al.*, 2015; Laloyaux *et al.*, 2015).

3.3 Objectifs

Le problème de couplage est souvent vu par la communauté d'assimilation de données comme un problème résolu. En effet, toutes les études détaillées précédemment présentent le modèle utilisé comme étant couplé, sans préciser ni de quel type de couplage il s'agit, ni s'il fournit une solution forte ou faible du problème du couplage (2.10) présenté dans le chapitre précédent. Il se trouve que l'intégralité des études présentées utilisent un modèle librement couplé. L'assimilation de données couplée nous semble être une opportunité pour développer de nouvelles méthodes permettant non seulement de prendre en compte le couplage des modèles, mais également de renforcer ce couplage grâce à l'assimilation pour proposer des analyses plus fortement couplées mathématiquement. L'objectif de cette thèse est donc de développer de nouvelles méthodes épaulant les modèles dans la recherche d'une solution au problème du couplage océan-atmosphère (2.10). Ces travaux s'articulent en deux parties :

- La première partie consiste à introduire de nouveaux algorithmes d'assimilation de données couplés. Nous ferons alors le lien avec des méthodes existantes, notamment l'algorithme CERA. Nous proposerons ensuite quelques résultats théoriques de convergence de ces algorithmes avant de réaliser des expériences numériques avec un modèle linéaire 1D couplé.
- La deuxième partie de ces travaux consiste à éprouver la robustesse de nos algorithmes en utilisant cette fois-ci un modèle 1D couplé océan-atmosphère plus réaliste, c'est à dire avec la présence de paramétrisations sous maille proche de l'interface. Ce modèle sera intégré à une surcouche logicielle OOPS, développée au CEPMMT, que nous présentons également.

Nous pourrons enfin conclure sur les performances de nos algorithmes et proposer quelques perspectives d'amélioration.

Synthèse des notations

Nous avons pu introduire dans cette première partie de nombreuses notations pour le couplage de modèles et l'assimilation de données. Par souci de clarté, nous proposons de résumer l'ensemble de ces notations dans les tableaux 3.2, 3.3, 3.4, 3.5 et 3.6.

Notation	Description
\cdot_a	indice désignant une composante atmosphérique
\cdot_o	indice désignant une composante océanique
$[0, T]$	Fenêtre temporelle
t	temps
\cdot_i	indice temporel
N	nombre de pas de temps
Δ_t	durée d'un pas de temps
N_j	Nombre de fenêtres temporelles
T_j	Fenêtre temporelle $[t_j, t_{j+1}]$ telle que $[0, T] = \bigcup_{j=1}^{N_j} T_j$
\mathcal{M}	Opérateur modèle continu
M	Modèle discrétisé non-linéaire
\mathbf{M}	Modèle linéaire
\mathbf{M}^T	Modèle adjoint
\mathbf{x}	Vecteur d'état
$\mathbf{x}_i = M_{0,i} \mathbf{x}_0$	Intégration du modèle du pas du temps $t = 0$ à $t = t_i$ \mathbf{x}_i représente le vecteur d'état au temps t_i

Tableau 3.2 – Tableau récapitulatif des notations temporelles, du modèle et de son vecteur d'état.

Notation	Description
\mathbf{x}^t	État vrai
$\boldsymbol{\eta}_i$	Erreur du vecteur d'état \mathbf{x}_i par rapport à l'état vrai au temps t_i
\mathbf{Q}_i	Matrice de covariance d'erreur de l'état (Erreur modèle)
\mathbf{x}^b	Ébauche
$\boldsymbol{\varepsilon}^b$	Erreur de l'ébauche par rapport à l'état vrai
\mathbf{B}	Matrice de covariance d'erreur de l'ébauche
\mathbf{x}^a	Analyse
$\boldsymbol{\varepsilon}^a$	Erreur de l'analyse par rapport à l'état vrai
\mathbf{A}	Matrice de covariance d'erreur de l'analyse
\mathbf{y}_i^o	Observation au temps t_i
$\boldsymbol{\varepsilon}_i^o$	Erreur de l'observation au temps t_i par rapport à l'état vrai au même instant
\mathbf{R}_i	Matrice de covariance d'erreur de l'observation au temps t_i
H_i	Opérateur d'observation au temps t_i
\mathbf{H}_i	Opérateur d'observation linéarisé
\mathbf{H}_i^T	Opérateur d'observation adjoint

Tableau 3.3 – Tableau récapitulatif des notations des différents vecteurs et matrices relatifs à l'assimilation de données.

Notation	Description
\mathbf{K}	Gain de Kalman
N_m	Nombre de membres d'un ensemble (dans l'EnKF par exemple)
m	Indice d'un membre d'un ensemble
J	Fonction coût à minimiser dans une méthode d'assimilation variationnelle
J^b	Terme de la fonction coût relatif à l'ébauche
J^o	Terme de la fonction coût relatif aux observations
∇J	Gradient de la fonction coût
$\nabla J^b, \nabla J^o$	Gradients relatifs à l'ébauche et aux observations
$\nabla^2 J$	Hessienne de la fonction coût
$\delta \mathbf{x}$	Incrément (dans le cadre du 4D-Var incrémental)
d_i	Écart de l'état du modèle par rapport aux observations au temps t_i

Tableau 3.4 – Tableau récapitulatif des notations intervenant dans les algorithmes d'assimilation de données stochastiques et variationnelles.

Notation	Description
Ω_a	Domaine atmosphérique
Ω_o	Domaine océanique
Γ	Interface air-mer
CLA	Couche limite atmosphérique
z_{CLA}	Hauteur de la CLA
CLS	Couche limite de surface
CLSA, CLSO	Couche limite de surface atmosphérique, océanique
z^+	Hauteur de la CLSA
z^-	Profondeur de la CLSO
CLO	Couche limite océanique
z_{CLO}	Profondeur de la CLO

Tableau 3.5 – Tableau récapitulatif des notations permettant de décrire le système océan-atmosphère.

Notation	Description
\mathbf{x}	Vecteur d'état du modèle
$(u, v, w)^T$	Composante de la quantité de mouvement de l'air ou de l'eau
\mathcal{F}	Traceur atmosphérique ou océanique (température, salinité...)
f_a, f_o	Termes sources des équations du modèle atmosphérique et océanique
$\mathcal{D}_a, \mathcal{D}_o$	Opérateurs d'interface atmosphériques et océaniques continus
D_a, D_o	Opérateurs d'interface atmosphériques et océaniques discrets
\mathcal{F}_{oa}	Flux à l'interface air-mer (continu)
F_{oa}	Flux à l'interface air-mer (discret)
\mathcal{R}	Flux radiatifs
C_x	Coefficient de transfert Bulk
f	Paramètre de Coriolis
g	Accélération de pesanteur
ρ	Masse volumique du fluide
K_m, K_s	Coefficients de viscosité et diffusivité turbulente
ν_m, ν_s	Coefficients de viscosité et de diffusion moléculaire
$\bar{\mathbf{x}}^j$	Champ moyenné sur la fenêtre T_j
\mathbf{x}^k	Champ à l'itération k de l'algorithme de Schwarz
$\mathbf{x}_{a \Gamma}$	Restriction de la composante atm. du vecteur d'état à l'interface Γ
$\mathbf{x}_{o \Gamma}$	Restriction de la composante océ. du vecteur d'état à l'interface Γ
$\mathbf{x}_\Gamma = (\mathbf{x}_{a \Gamma}, \mathbf{x}_{o \Gamma})^T$	Restriction du vecteur d'état à l'interface Γ

Tableau 3.6 – Tableau récapitulatif des notations permettant de décrire les équations des modèles d'océan et d'atmosphère ainsi que leur couplage.

Deuxième partie

ÉTUDE ET DÉVELOPPEMENT D'ALGORITHMES D'ASSIMILATION DE DONNÉES VARIATIONNELLE COUPLÉE

Nous venons d'introduire dans la première partie le problème de l'assimilation de données couplée ainsi que des solutions proposées dans diverses études depuis les années 1990. Dans cette seconde partie, nous proposerons des alternatives à ces méthodes existantes pour répondre au problème. Nous réaliserons également une étude mathématique et numérique de ces algorithmes. Cette partie est divisée en deux chapitres :

- Le chapitre 4 est consacré à la présentation de quelques méthodes d'assimilation de données couplée que nous proposons comme alternative aux algorithmes existants. Nous introduirons alors une nouvelle classification des algorithmes variationnels d'assimilation couplée, à savoir les méthodes homogènes et inhomogènes. Ces deux types de méthodes se distingueront dans le choix du modèle linéaire au sein de la boucle interne du 4D-Var incrémental. Nous proposerons finalement quelques résultats théoriques de convergence de ces algorithmes.
 - Le chapitre 5 concerne l'étude numérique des algorithmes introduits dans le chapitre 4. Nous nous placerons dans un cadre très simple avec l'utilisation d'un modèle linéaire couplé. Le but est ici de présenter une première validation numérique de ces méthodes avant de passer à un cadre d'application plus complexe dans la troisième et dernière partie de ce manuscrit.
-

Chapitre 4

Introduction de quelques méthodes d'assimilation couplée

Sommaire

4.1 Méthodes d'assimilation de données couplée	64
4.1.1 Algorithmes <i>homogènes</i>	64
4.1.2 Algorithmes <i>inhomogènes</i>	70
4.2 Étude de convergence des algorithmes d'assimilation couplée	74
4.2.1 Le 4D-Var incrémental vu comme un algorithme de Gauss-Newton	74
4.2.2 Convergence des algorithmes homogènes	76
4.2.3 Convergence des algorithmes inhomogènes	76

Nous proposons dans ce chapitre de décrire plusieurs méthodes d'assimilation de données couplée variationnelles, c'est à dire cherchant à produire une analyse qui soit solution du problème de couplage océan-atmosphère (2.9). Ces méthodes seront toute basées sur le 4D-Var incrémental précédemment introduit. Rappelons que le problème du couplage consiste à trouver $\mathbf{x} = (\mathbf{x}_a, \mathbf{x}_o)^T$ tel que :

$$\left\{ \begin{array}{ll} M_{a,0,T}(\mathbf{x}_{a,0}) = f_a & \text{sur } \Omega_a \times [0, T] \\ \mathbf{x}_{a,0}(z) = \mathbf{x}_0(z) & z \in \Omega_a \\ D_a(\mathbf{x}_{a|_\Gamma}) = F_{oa}(\mathbf{x}_{a|_\Gamma}, \mathbf{x}_{o|_\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \end{array} \right. \left\{ \begin{array}{ll} M_{o,0,T}(\mathbf{x}_{o,0}) = f_o & \text{sur } \Omega_o \times [0, T] \\ \mathbf{x}_{o,0}(z) = \mathbf{x}_0(z) & z \in \Omega_o \\ D_o(\mathbf{x}_{o|_\Gamma}) = F_{oa}(\mathbf{x}_{a|_\Gamma}, \mathbf{x}_{o|_\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \end{array} \right.$$

où $\mathbf{x}_0(z) \in \Omega$ désigne la condition initiale. Les deux stratégies de couplage que nous considérerons dans ce manuscrit sont le couplage asynchrone et par itérations de Schwarz. Soit la décomposition de la fenêtre temporelle $[0, T]$ en N_j sous-fenêtres $T_j = [t_j, t_{j+1}]$ telles que $[0, T] = \bigcup_{j=0}^{N_j-1} T_j$, nous pouvons alors écrire ces méthodes de couplage sous un formalisme commun :

$$\left\{ \begin{array}{ll} M_{a,j,j+1}(\mathbf{x}_{a,j}^k) = f_a & \text{sur } \Omega_a \times T_j \\ \mathbf{x}_{a,j}^k(z) = \mathbf{x}_j^k(z) & z \in \Omega_a \\ D_a(\mathbf{x}_{a|_\Gamma}^k) = F_{oa}(\mathbf{x}_{a|_\Gamma}^k, \mathbf{x}_{o|_\Gamma}^{k-1}, \mathcal{R}) & \text{sur } \Gamma \times T_j \end{array} \right. \left\{ \begin{array}{ll} M_{o,j,j+1}(\mathbf{x}_{o,j}^k) = f_o & \text{sur } \Omega_o \times T_j \\ \mathbf{x}_{o,j}^k(z) = \mathbf{x}_j^k(z) & z \in \Omega_o \\ D_o(\mathbf{x}_{o|_\Gamma}^k) = F_{oa}(\mathbf{x}_{a|_\Gamma}^k, \mathbf{x}_{o|_\Gamma}^k, \mathcal{R}) & \text{sur } \Gamma \times T_j \end{array} \right.$$

où $\mathbf{x}_j(z)$ et le first-guess $\mathbf{x}_{o|_\Gamma}^0$ sont imposés sur T_0 puis fournis par la solution du modèle sur la fenêtre temporelle précédente sur T_j avec $j > 0$. Nous pouvons définir un nombre d'itérations maximal k_{\max} tel que :

- si $k_{\max} = 1$, nous avons un couplage asynchrone,
- si $k_{\max} > 1$, nous avons un couplage itératif de Schwarz. Si k_{\max} est suffisamment grand, alors nous aurons une solution forte au problème du couplage.

Considérant ces deux stratégies de couplage, nous allons pouvoir introduire plusieurs méthodes d'assimilation de données couplée que nous classerons en deux grandes catégories. D'une part nous aurons les algorithmes *homogènes* et d'autre part les algorithmes *inhomogènes*. Nous entendons par *homogènes* les algorithmes où la fonction coût minimisée dans la boucle interne est la stricte linéarisation de la fonction coût de la boucle externe. A l'inverse, les méthodes *inhomogènes* considèrent un problème différent à minimiser entre la boucle interne et externe. Nous pourrons alors faire le lien entre nos algorithmes et le système CERA qui se trouve être une méthode inhomogène. Enfin nous terminerons ce chapitre par une brève étude de convergence des algorithmes introduits.

Remarques

Nous considérerons dans ce chapitre que :

- $\mathbf{B} = \begin{pmatrix} \mathbf{B}_a & 0 \\ 0 & \mathbf{B}_o \end{pmatrix}$ est la matrice de covariance d'erreur d'ébauche couplée avec \mathbf{B}_a et \mathbf{B}_o les matrices de covariance d'erreurs atmosphériques et océaniques. Nous ne considérerons pas de covariances croisées.
- $\mathbf{R} = \begin{pmatrix} \mathbf{R}_a & 0 \\ 0 & \mathbf{R}_o \end{pmatrix}$ est la matrice de covariance d'erreur d'observations couplée suivant le même principe que \mathbf{B} .
- $H = \begin{pmatrix} H_a & 0 \\ 0 & H_o \end{pmatrix}$ est l'opérateur d'observation.
- M désigne le modèle couplé, de manière asynchrone ou par itérations de Schwarz, et M_a et M_o désignent respectivement les modèles atmosphériques et océaniques.

4.1 Méthodes d'assimilation de données couplée

4.1.1 Algorithmes *homogènes*

Cette première famille d'algorithmes que nous présentons correspond à ceux où la fonction coût linéaire J de la boucle interne est la stricte linéarisation de la fonction coût J de la boucle externe. Cela signifie que le modèle linéaire et adjoint seront couplés si le modèle non-linéaire l'est également. Dans la littérature, ces algorithmes sont souvent désignés comme *fortement couplés* (par exemple dans le papier de Smith *et al.* (2015)).

Méthode Homogène Couplée (MH-C)

La première approche présentée consiste simplement à réaliser un 4D-Var incrémental avec un modèle couplé non-linéaire, sa linéarisation et son adjoint correspondant. Le vecteur de contrôle est l'état initial du système $\mathbf{x}_0 \in \Omega$ et la fonction coût à minimiser est :

$$J_{\text{MH-C}}(\mathbf{x}_0^{(p)}) = \frac{1}{2} (\mathbf{x}_0^{(p)} - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0^{(p)} - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N \left(H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right)^T \mathbf{R}_i^{-1} \left(H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right)$$

où p dénote les itérations de boucle externes et $\mathbf{x}_0^{(p)}$ est l'état initial du système évalué à l'itération p du 4D-Var incrémental. Cette méthode produira une analyse aussi couplée que le modèle. Par exemple, si le modèle est librement couplé avec une méthode asynchrone, alors l'analyse produite sera librement couplée. Concernant son implémentation incrémentale, le MH-C évalue la trajectoire du modèle à partir du modèle couplé pour ensuite définir les écarts aux observations :

$$d_i^{(p)} = H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \quad \forall i \in \llbracket 0, N \rrbracket$$

puis, la fonction coût linéaire à minimiser et son gradient sont calculés à partir du modèle linéaire couplé \mathbf{M} et de son adjoint couplé \mathbf{M}^T :

$$J_{\text{MH-C}}(\delta \mathbf{x}_0^{(p)}) = \frac{1}{2} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right)^T \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) + \frac{1}{2} \sum_{i=0}^N (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)})^T \mathbf{R}_i^{-1} (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)})$$

$$\nabla J_{\text{MH-C}}(\delta \mathbf{x}_0^{(p)}) = \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) - \sum_{i=0}^N \mathbf{M}_1^T \dots \mathbf{M}_i^T \mathbf{H}_i^T \mathbf{R}_i^{-1} (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_i \dots \mathbf{M}_1 \delta \mathbf{x}_0^{(p)})$$

Une représentation schématique du MH-C est donnée par la figure 4.1. La force de cette méthode est qu'elle est capable de produire une analyse fortement couplée si le modèle l'est également, par exemple avec un couplage de type Schwarz. Cependant, il s'agit également de sa principale limitation. En effet, un algorithme de Schwarz peut nécessiter de nombreuses itérations pour arriver à convergence. Ces itérations, jointes aux itérations du 4D-Var incrémental, peuvent augmenter considérablement le coût numérique de l'algorithme, le rendant alors inutilisable en pratique. De plus, cette méthode nécessite de développer les modèles linéaire et adjoint du modèle couplé, ce qui peut s'avérer complexe. Nous pourrions considérer des variantes de cet algorithme selon le choix du vecteur de contrôle :

- contrôle de la seule condition initiale $\mathbf{x}_0 \in \Omega$: **MH-C**,
- contrôle de la condition initiale et du first-guess à l'interface $\mathbf{x}_{0|\Gamma}^0 \in \Gamma \times T_0$: **MH-C- Γ** ,
- contrôle de la condition initiale et d'un terme de correction des champs à l'interface $\delta \mathbf{x}_\Gamma \in \Gamma \times [0, T]$: **MH-C- $\delta\Gamma$** .



Méthode Homogène Couplée (MH-C)

- 4D-Var incrémental avec utilisation d'un modèle couplé dans la boucle externe
- Modèle linéaire couplé et adjoint couplé dans la boucle interne
- + Analyse aussi couplée que le modèle
- Développement du modèle adjoint couplé
- Coût de calcul très élevé possible selon le type de couplage utilisé

Méthode Homogène Couplée Renforcée (MH-CR)

Puisque le MH-C est trop coûteux numériquement avec un modèle fortement couplé, un modèle faiblement ou librement couplé devrait être utilisé en pratique dans cet algorithme. Or,

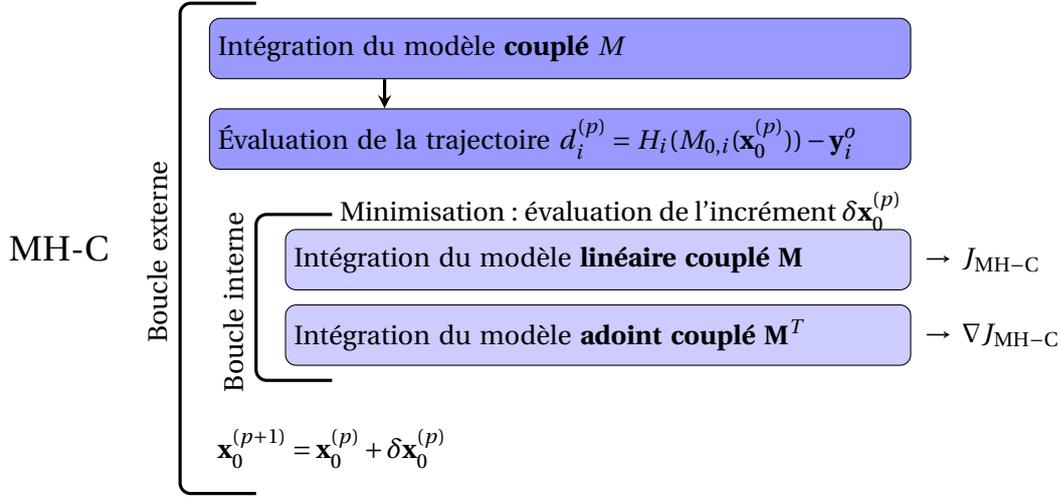


FIGURE 4.1 – Représentation schématique de l'algorithme MH-C basé sur le 4D-Var incrémental. Le modèle couplé est utilisé dans la boucle externe pour évaluer la trajectoire ainsi que dans la boucle interne pour la minimisation.

l'utilisation d'un tel modèle conduirait à la création d'une analyse faiblement couplée mathématiquement, comme nous l'avons mentionné. Nous proposons donc un second algorithme visant à *renforcer* le couplage lors de la création de l'analyse, le MH-CR. L'idée de cet algorithme provient des travaux de Gejadze et Monnier (2007) qui introduisent un nouveau terme de pénalisation dans la fonction coût J du 4D-var. Ce terme a pour but de pénaliser les erreurs de couplage à l'interface. Leur étude a été menée dans le cadre de l'hydraulique fluviale avec deux modèles 2D et 1D non linéaires d'équations de Saint-Venant. Ils cherchent à évaluer de meilleures conditions aux frontières du modèle 2D, imbriqué dans le modèle 1D, via une méthode de contrôle optimal. Nous avons souhaité adapter cette méthode dans le cas de deux modèles couplés d'océan et d'atmosphère. Considérons un modèle librement couplé par une méthode asynchrone par exemple. La solution proposée par le modèle ne respecte donc pas la consistance des flux à l'interface air-mer. Le principe du MH-CR est alors de minimiser l'erreur de couplage commise par le modèle au cours de l'assimilation grâce à un terme supplémentaire dans la fonction coût J^s tel que :

$$J^s(\mathbf{x}_0) = \frac{1}{2} \sum_{i=0}^N \left(D_a(\mathbf{x}_{a|\Gamma,i}^{k_{\max}}) - D_o(\mathbf{x}_{o|\Gamma,i}^{k_{\max}}) \right)^T \mathbf{W}^{-1} \left(D_a(\mathbf{x}_{a|\Gamma,i}^{k_{\max}}) - D_o(\mathbf{x}_{o|\Gamma,i}^{k_{\max}}) \right) \quad (4.1)$$

où \mathbf{W} est une matrice permettant de différencier les poids des différents flux d'interface. $\mathbf{x}_{\alpha,i}^{k_{\max}}$ ($\alpha = a, o$) fait référence à la solution du modèle M après k_{\max} itérations, et D_α sont les opérateurs d'interface. La fonction coût totale du MH-CR est alors :

$$J_{MH-CR}(\mathbf{x}_0^{(p)}) = J^b(\mathbf{x}_0^{(p)}) + \frac{1}{2} \sum_{i=0}^N \left(H_i(M_{0,i}^{k_{\max}}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right)^T \mathbf{R}_i^{-1} \left(H_i(M_{0,i}^{k_{\max}}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right) + \gamma J^s(\mathbf{x}_0^{(p)})$$

avec $M_{0,i}^{k_{\max}}(\mathbf{x}_0) = \left(\mathbf{x}_{a,i}^{k_{\max}}, \mathbf{x}_{o,i}^{k_{\max}} \right)^T$ et γ est un coefficient permettant de jouer sur le poids global de J^s dans la fonction coût. Pour minimiser cette fonction coût, le MH-CR évalue la trajectoire dans la boucle externe à partir du modèle couplé. Les écarts aux observations d_i et l'erreur de

couplage ε_i^c du modèle non-linéaire sont alors suivants :

$$\begin{aligned} d_i^{(p)} &= H_i(M_{0,i}^{k_{\max}}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o & \forall i \in \llbracket 0, N \rrbracket \\ \varepsilon_i^{c,(p)} &= D_a^\Omega(M_{0,i}^{k_{\max}}(\mathbf{x}_0^{(p)})) - D_o^\Omega(M_{0,i}^{k_{\max}}(\mathbf{x}_0^{(p)})) & \forall i \in \llbracket 0, N \rrbracket \end{aligned}$$

Remarque 4.1.1

Les opérateurs D_a^Ω sont des extensions des opérateurs D_a sur le domaine Ω . C'est à dire que $D_a : \Gamma \rightarrow \Gamma$ et $D_a^\Omega : \Omega \rightarrow \Gamma$, mais le calcul des flux à l'interface est identique pour les deux opérateurs. Nous avons besoin d'introduire cette notation afin de pouvoir écrire la fonction coût en fonction de \mathbf{x}_0 et du modèle, et de nous passer de la notation $\mathbf{x}_{\alpha|\Gamma}$.

Le modèle linéaire couplé $\mathbf{M}^{k_{\max}}$ et son adjoint couplé $\mathbf{M}^{k_{\max}T}$ sont ensuite utilisés dans la boucle interne pour minimiser la fonction coût linéaire :

$$\begin{aligned} J_{\text{MH-CR}}(\delta \mathbf{x}_0^{(p)}) &= \frac{1}{2} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right)^T \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) + \\ &\quad \frac{1}{2} \sum_{i=0}^N (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)})^T \mathbf{R}_i^{-1} (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)}) + \\ &\quad \gamma \frac{1}{2} \sum_{i=0}^N \left(\mathbf{D}_a^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} - \mathbf{D}_o^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} + \varepsilon_i^{c,(p)} \right)^T \mathbf{W}^{-1} \\ &\quad \left(\mathbf{D}_a^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} - \mathbf{D}_o^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} + \varepsilon_i^{c,(p)} \right) \\ \nabla J_{\text{MH-CR}}(\delta \mathbf{x}_0^{(p)}) &= \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) - \sum_{i=0}^N \mathbf{M}_{0,i}^{k_{\max}T} \mathbf{H}_i^T \mathbf{R}_i^{-1} (d_i^{(p)} - \mathbf{H}_i \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)}) \\ &\quad + \gamma \sum_{i=0}^N \mathbf{M}_{0,i}^{k_{\max}T} \mathbf{D}_a^{\Omega T} \mathbf{W}^{-1} \left(\mathbf{D}_a^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} - \mathbf{D}_o^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} + \varepsilon_i^{c,(p)} \right) \\ &\quad - \gamma \sum_{i=0}^N \mathbf{M}_{0,i}^{k_{\max}T} \mathbf{D}_o^{\Omega T} \mathbf{W}^{-1} \left(\mathbf{D}_a^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} - \mathbf{D}_o^\Omega \mathbf{M}_{0,i}^{k_{\max}} \delta \mathbf{x}_0^{(p)} + \varepsilon_i^{c,(p)} \right) \end{aligned}$$

Nous proposons une représentation schématique de cet algorithme dans la figure 4.2. Nous espérons renforcer le couplage entre les deux modèles grâce à la contrainte faible de couplage imposée par J^s , à la manière d'un algorithme de Schwarz. Les itérations de minimisation prennent alors le relais sur les itérations de Schwarz. Le MH-CR présente cependant la même limitation que le MH-C, à savoir la nécessité de développer le modèle adjoint couplé. Plusieurs variantes sont envisageables, en fonction du choix du vecteur de contrôle :

- contrôle de la seule condition initiale $\mathbf{x}_0 \in \Omega$: **MH-CR**,
- contrôle de la condition initiale et du first-guess à l'interface $\mathbf{x}_{0|\Gamma}^0 \in \Gamma \times T_0$: **MH-CR- Γ** ,
- contrôle de la condition initiale et d'un terme de correction des champs à l'interface $\delta \mathbf{x}_\Gamma \in \Gamma \times [0, T]$: **MH-CR- $\delta\Gamma$** .



Méthode Homogène Couplée Renforcée (MH-CR)

- 4D-Var incrémental avec utilisation d'un modèle faiblement couplé dans la boucle externe
- Modèle linéaire faiblement couplé et adjoint couplé dans la boucle interne
 - + Analyse couplée grâce au modèle et à la contrainte faible imposée par J^s
 - + Coût de calcul potentiellement plus faible que le MH-C
 - Développement du modèle adjoint couplé

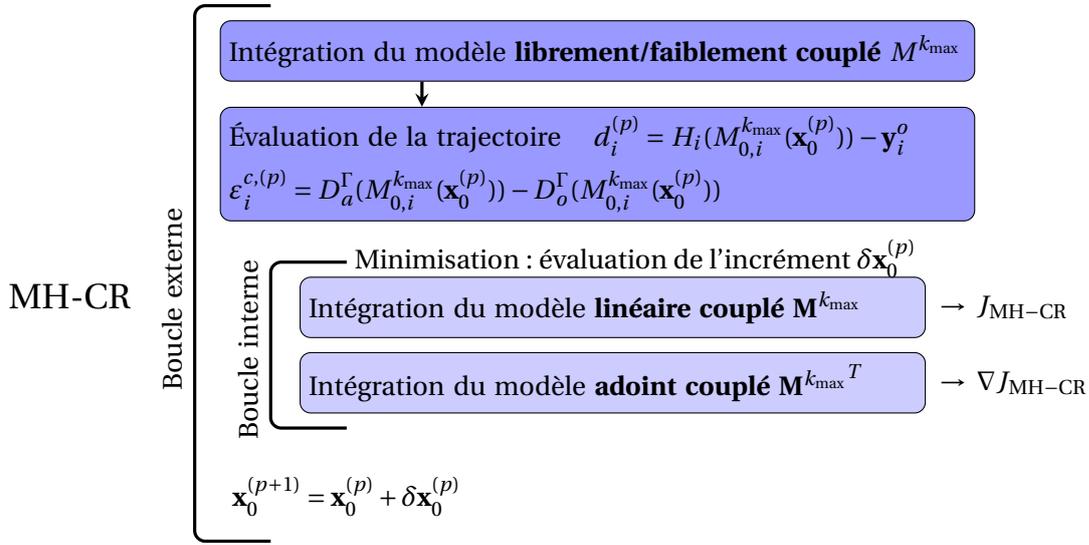


FIGURE 4.2 – Représentation schématique de l’algorithme MH-CR basé sur le 4D-Var incrémental. Le modèle couplé est utilisé dans la boucle externe pour évaluer la trajectoire ainsi que dans la boucle interne pour la minimisation. L’originalité de cet algorithme vient du terme de pénalisation des inconsistances de flux à l’interface Γ dans la fonction coût.

Méthode Homogène Couplée par Contrainte Faible (MH-CCF)

Enfin, nous présentons une dernière méthode homogène qui consiste à assurer le couplage entre les deux modèles par le seul terme de pénalisation J^s . Les deux modèles M_a et M_o sont découplés dans la boucle externe ainsi que leurs composantes linéaires et adjointes dans la boucle interne. La fonction coût de cette méthode est alors :

$$\begin{aligned}
 J_{\text{MH-CCF}}(\mathbf{x}_0^{(p)}) &= J_a^b(\mathbf{x}_{a,0}^{(p)}) + J_o^b(\mathbf{x}_{o,0}^{(p)}) \\
 &\quad + \frac{1}{2} \sum_{i=0}^N \left(H_{a,i}(M_{a,0,i}(\mathbf{x}_{a,0}^{(p)})) - \mathbf{y}_{a,i}^o \right)^T \mathbf{R}_{a,i}^{-1} \left(H_{a,i}(M_{a,0,i}(\mathbf{x}_{a,0}^{(p)})) - \mathbf{y}_{a,i}^o \right) \\
 &\quad + \frac{1}{2} \sum_{i=0}^N \left(H_{o,i}(M_{o,0,i}(\mathbf{x}_{o,0}^{(p)})) - \mathbf{y}_{o,i}^o \right)^T \mathbf{R}_{o,i}^{-1} \left(H_{o,i}(M_{o,0,i}(\mathbf{x}_{o,0}^{(p)})) - \mathbf{y}_{o,i}^o \right) \\
 &\quad + \gamma J^s(\mathbf{x}_0^{(p)}) \\
 &= J_a^b(\mathbf{x}_{a,0}^{(p)}) + J_o^b(\mathbf{x}_{o,0}^{(p)}) + J_a^o(\mathbf{x}_{a,0}^{(p)}) + J_o^o(\mathbf{x}_{o,0}^{(p)}) + \gamma J^s(\mathbf{x}_0^{(p)})
 \end{aligned}$$

où J_a^b (respectivement J_o^b) est le terme d'ébauche pour la composante atmosphérique (resp. océanique) du vecteur d'état et :

$$J^s(\mathbf{x}_0^{(p)}) = \frac{1}{2} \sum_{i=0}^N \left(D_a^{\Omega_a}(M_{a,0,i}(\mathbf{x}_{a,0}^{(p)})) - D_o^{\Omega_o}(M_{o,0,i}(\mathbf{x}_{o,0}^{(p)})) \right)^T \mathbf{W}^{-1} \left(D_a^{\Omega_a}(M_{a,0,i}(\mathbf{x}_{a,0}^{(p)})) - D_o^{\Omega_o}(M_{o,0,i}(\mathbf{x}_{o,0}^{(p)})) \right)$$

Remarque 4.1.2

De manière analogue à précédemment, nous définissons ici les opérateurs $D_a^{\Omega_a}$ comme les extensions des opérateurs D_a sur le domaine Ω_a . Nous avons alors $D_a^{\Omega_a} : \Omega_a \rightarrow \Gamma$.

La figure 4.3 représente schématiquement cet algorithme. La qualité du couplage sera dégradée par rapport à un couplage direct des modèles, mais cet algorithme présente l'avantage de ne pas avoir de couplage direct ou adjoint des modèles. Il est également possible de définir plusieurs variantes du MH-CCF :

- contrôle de la seule condition initiale $\mathbf{x}_0 \in \Omega$: **MH-CCF**,
- contrôle de la condition initiale et de l'interface $\mathbf{x}_\Gamma \in \Gamma \times T_0$: **MH-CCF- Γ** .

Remarque 4.1.3

Cet algorithme MH-CCF est très similaire à la méthode ADMM (*Alternative Direction Method of Multipliers*), introduite par Glowinski et Marroco (1975) puis Gabay et Mercier (1976), et qui consiste à résoudre un problème de la forme :

$$\begin{aligned} & \text{minimiser} && J_1(\mathbf{x}_1) + J_2(\mathbf{x}_2) \\ & \text{tel que} && C_1\mathbf{x}_1 + C_2\mathbf{x}_2 = \mathbf{c} \end{aligned}$$

où \mathbf{x}_1 , \mathbf{x}_2 et \mathbf{c} sont des vecteurs, J_1 et J_2 sont des fonctions coût convexes et C_1 et C_2 des opérateurs. Le MH-CCF est donc équivalent à l'ADMM dans le cas particulier où les modèles d'atmosphère et d'océan sont linéaires. La contrainte $C_1\mathbf{x}_1 + C_2\mathbf{x}_2 = \mathbf{c}$ correspond alors à notre contrainte faible imposée par J^s et J_1 et J_2 correspondent aux fonctions coût relatives aux variables atmosphériques et océaniques ($J_a^{b,o}$ et $J_o^{b,o}$). Il existe plusieurs études de convergence de cette méthode dans la littérature (Gabay, 1983; Eckstein et Bertsekas, 1992).



Méthode Homogène Couplée par Contrainte Faible (MH-CCF)

- 4D-Var incrémental avec deux modèles découplés dans la boucle externe
- Modèles linéaires et adjoints découplés également dans la boucle interne
- Analyse faiblement couplée par la contrainte faible imposée par J^s
- + Coût de calcul raisonnable les modèles étant découplés
- + Pas de développement du modèle adjoint couplé

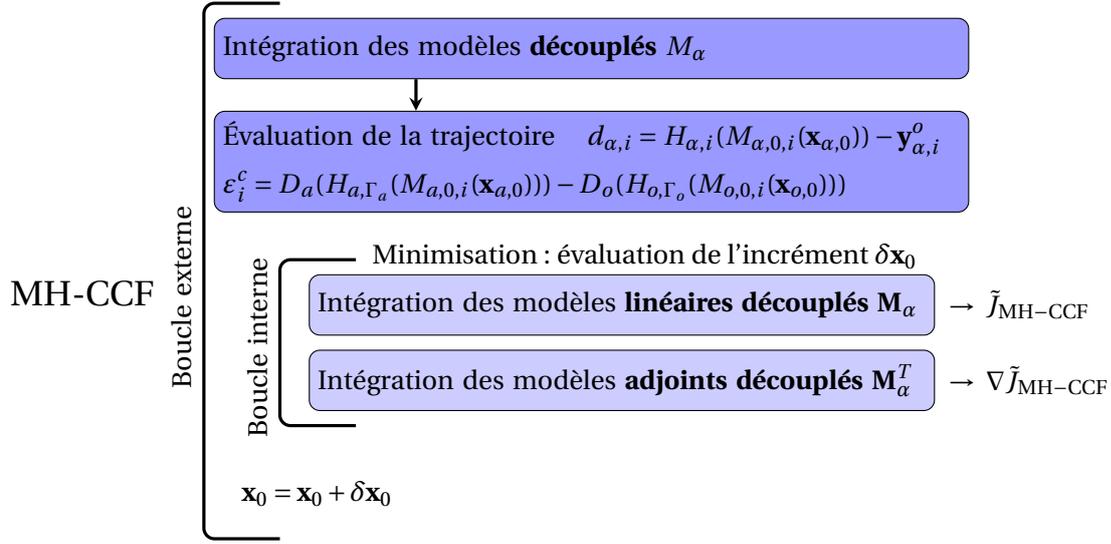


FIGURE 4.3 – Représentation schématique de l’algorithme MH-CCF basé sur le 4D-Var incrémental. Les modèles sont découplés mais l’analyse est couplée via la contrainte faible imposée par J^s durant la minimisation.

4.1.2 Algorithmes *inhomogènes*

Les méthodes introduites ci-avant présentent quelques limitations. Le MH-C ou le MH-CR nécessitent le développement de l’adjoint du couplage et peuvent être très gourmandes en coût de calcul. Le MH-CCF ne requiert pas l’adjoint du couplage mais l’absence d’un modèle direct couplé impacte la qualité de l’analyse à l’interface air-mer. C’est pourquoi il peut être judicieux de faire appel à des méthodes que nous appellerons *inhomogènes*, utilisant un modèle couplé dans la boucle externe du 4D-Var, et leurs composantes découplées dans la boucle interne. La fonction coût quadratique minimisée au sein de la boucle interne est alors une approximation de la vraie fonction coût quadratique (la stricte linéarisation de la fonction coût de la boucle externe).

Méthode Inhomogène Couplée (MI-C)

Cette méthode est l’équivalente du MH-C pour les algorithmes inhomogènes. Le principe est simplement de remplacer les modèles couplés linéaire et adjoint par les composantes découplées dans la boucle interne. Nous calculons alors deux incréments $\delta \mathbf{x}_{a,0}$ et $\delta \mathbf{x}_{o,0}$. Les fonctions coûts et gradient du MI-C sont les suivants :

$$J_{\text{MI-C}}(\mathbf{x}_0^{(p)}) = \frac{1}{2} \left(\mathbf{x}_0^{(p)} - \mathbf{x}_0^b \right)^T \mathbf{B}^{-1} \left(\mathbf{x}_0^{(p)} - \mathbf{x}_0^b \right) + \frac{1}{2} \sum_{i=0}^N \left(H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right)^T \mathbf{R}_i^{-1} \left(H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o \right)$$

$$\begin{aligned}
\tilde{J}_{\text{MI-C}}(\delta \mathbf{x}_0^{(p)}) &= \frac{1}{2} \left(\delta \mathbf{x}_{a,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{a,0}^{(l)} \right)^T \mathbf{B}_a^{-1} \left(\delta \mathbf{x}_{a,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{a,0}^{(l)} \right) + \\
&\quad \frac{1}{2} \sum_{i=0}^N \left(d_{a,i}^{(p)} - \mathbf{H}_{a,i} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} \right)^T \mathbf{R}_{a,i}^{-1} \left(d_{a,i}^{(p)} - \mathbf{H}_{a,i} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} \right) + \\
&\quad \frac{1}{2} \left(\delta \mathbf{x}_{o,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{o,0}^{(l)} \right)^T \mathbf{B}_o^{-1} \left(\delta \mathbf{x}_{o,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{o,0}^{(l)} \right) + \\
&\quad \frac{1}{2} \sum_{i=0}^N \left(d_{o,i}^{(p)} - \mathbf{H}_{o,i} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} \right)^T \mathbf{R}_{o,i}^{-1} \left(d_{o,i}^{(p)} - \mathbf{H}_{o,i} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} \right) \\
\nabla \tilde{J}_{\text{MI-C}}(\delta \mathbf{x}_0^{(p)}) &= \mathbf{B}_a^{-1} \left(\delta \mathbf{x}_{a,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{a,0}^{(l)} \right) - \\
&\quad \sum_{i=0}^N \mathbf{M}_{a,0,i}^T \mathbf{H}_{a,i}^T \mathbf{R}_{a,i}^{-1} \left(d_{a,i}^{(p)} - \mathbf{H}_{a,i} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} \right) + \\
&\quad \mathbf{B}_o^{-1} \left(\delta \mathbf{x}_{o,0}^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_{o,0}^{(l)} \right) - \\
&\quad \sum_{i=0}^N \mathbf{M}_{o,0,i}^T \mathbf{H}_{o,i}^T \mathbf{R}_{o,i}^{-1} \left(d_{o,i}^{(p)} - \mathbf{H}_{o,i} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} \right)
\end{aligned}$$

avec $d_i^{(p)} = H_i(M_{0,i}(\mathbf{x}_0^{(p)})) - \mathbf{y}_i^o = (d_{a,i}^{(p)}, d_{o,i}^{(p)})^T$ évalué à partir du modèle couplé. On dénote par $\tilde{\cdot}$ le fait que la fonction coût quadratique (et son gradient) sont une approximation de la linéarisation exacte de la fonction coût en boucle externe (et de son gradient). La figure 4.4 représente cet algorithme. On remarquera que le MI-C est l'équivalent du système CERA (Laloyaux *et al.*, 2015) dans le cas particulier où le modèle est couplé de manière asynchrone dans la boucle externe et que seulement deux itérations de boucles externes sont effectuées. Le MI-C présente l'avantage de produire une analyse couplée grâce au modèle couplé dans la boucle externe tout en évitant d'avoir à développer l'adjoint du couplage. On espère alors que les itérations de boucles externes permettront de créer des covariances croisées entre les deux domaines grâce à la dynamique du modèle direct couplé.

Remarque 4.1.4

La fonction coût non-linéaire à minimiser du MI-C est la même que celle du MH-C. Cependant, la convergence vers la solution du MH-C n'est pas assurée. En effet, le problème à résoudre entre la boucle interne et externe étant différent, rien ne nous assure a priori que la solution trouvée sera celle de la boucle externe. Nous étudierons dans la suite de ce chapitre la convergence du MI-C avec les itérations de boucles externes.



Méthode Inhomogène Couplée (MI-C)

- 4D-Var incrémental avec un modèle couplé dans la boucle externe
- Modèles linéaires et adjoints découplés dans la boucle interne
 - Problèmes différents entre la boucle interne et externe
 - + Coût de calcul raisonnable les modèles étant découplés dans la boucle interne
 - + Pas de développement du modèle adjoint couplé

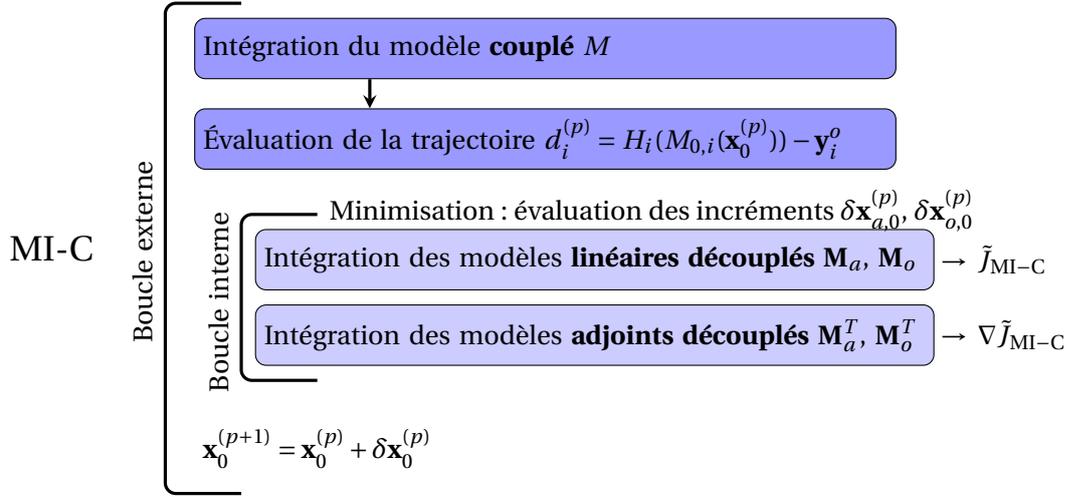


FIGURE 4.4 – Représentation schématique de l’algorithme MI-C basé sur le 4D-Var incrémental. Le modèle couplé est utilisé dans la boucle externe pour évaluer la trajectoire et la minimisation est réalisée séparément avec les deux modèles découplés.

Méthode Inhomogène Couplée Renforcée par Contrainte Faible (MI-CRCF)

Enfin, un dernier algorithme considéré renforce le couplage des deux modèles découplés dans la boucle interne via la contrainte faible imposée par J^s . Nous avons les fonctions coût suivantes :

$$J_{\text{MI-CRCF}}(\mathbf{x}_0^{(p)}) = J_{\text{MI-C}}(\mathbf{x}_0^{(p)}) + J^s(\mathbf{x}_0^{(p)})$$

avec

$$J^s(\mathbf{x}_0^{(p)}) = \frac{1}{2} \gamma \sum_{i=0}^N \left(D_a^\Omega(M_{0,i}(\mathbf{x}_0^{(p)})) - D_o^\Omega(M_{0,i}(\mathbf{x}_0^{(p)})) \right)^T \mathbf{W}^{-1} \left(D_a^\Omega(M_{0,i}(\mathbf{x}_0^{(p)})) - D_o^\Omega(M_{0,i}(\mathbf{x}_0^{(p)})) \right)$$

Nous considérons alors la fonction coût linéaire suivante au sein de la boucle interne :

$$\tilde{J}_{\text{MI-CRCF}}(\delta \mathbf{x}_0^{(p)}) = \tilde{J}_{\text{MI-C}}(\delta \mathbf{x}_0^{(p)}) + \frac{1}{2} \sum_{i=0}^N \left(\sqrt{\tilde{\gamma}} \mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} - \sqrt{\tilde{\gamma}} \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} + \sqrt{\tilde{\gamma}} \varepsilon_i^{c,(p)} \right)^T \mathbf{W}^{-1} \left(\sqrt{\tilde{\gamma}} \mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} - \sqrt{\tilde{\gamma}} \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} + \sqrt{\tilde{\gamma}} \varepsilon_i^{c,(p)} \right)$$

où ε_i^c correspond à l’erreur de couplage commise par le modèle non-linéaire, évaluée en boucle externe. Nous nous laissons également le choix de considérer un facteur $\tilde{\gamma}$ différent de γ . Nous pouvons alors écrire le gradient de la fonction coût linéaire :

$$\begin{aligned} \nabla \tilde{J}_{\text{MI-CRCF}}(\delta \mathbf{x}_0^{(p)}) &= \nabla \tilde{J}_{\text{MI-C}}(\delta \mathbf{x}_0^{(p)}) + \\ &+ \tilde{\gamma} \sum_{i=0}^N \mathbf{M}_{a,0,i}^T \mathbf{D}_a^{\Omega_a T} \mathbf{W}^{-1} \left(\mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} - \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} + \varepsilon_i^{c,(p)} \right) \\ &- \tilde{\gamma} \sum_{i=0}^N \mathbf{M}_{o,0,i}^T \mathbf{D}_o^{\Omega_o T} \mathbf{W}^{-1} \left(\mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta \mathbf{x}_{a,0}^{(p)} - \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta \mathbf{x}_{o,0}^{(p)} + \varepsilon_i^{c,(p)} \right) \end{aligned}$$

Cet algorithme, schématisé dans la figure 4.5, a pour but de renforcer le couplage au sein de l’analyse grâce à la contrainte faible J^s par rapport à l’analyse produite par le MI-C. Cette

contrainte permet alors d'assurer un couplage entre les modèles tout en évitant de devoir développer l'adjoint du modèle couplé. On espère ainsi améliorer la convergence de l'algorithme vers la solution du MH-C. Tout comme le MH-CR et le MH-CCF, nous pouvons imaginer cet algorithme en considérant différents choix de vecteur de contrôle :

- contrôle de la seule condition initiale $\mathbf{x}_0 \in \Omega$: **MI-CRCF**,
- contrôle de la condition initiale et du first-guess à l'interface $\mathbf{x}_{o|f}^0 \in \Gamma \times T_0$: **MI-CRCF- Γ** ,
- contrôle de la condition initiale et d'un terme de correction des champs à l'interface $\delta \mathbf{x}_\Gamma \in \Gamma \times [0, T]$: **MI-CRCF- $\delta\Gamma$** .

L'ensemble des algorithmes présentés dans ce chapitre est récapitulé dans le tableau 4.1.

 **Méthode Inhomogène Couplée Renforcée par Contrainte Faible (MI-CRCF)**

- 4D-Var incrémental avec un modèle couplé dans la boucle externe
- Modèles linéaires et adjoints découplés dans la boucle interne
 - Problèmes différents entre la boucle interne et externe
 - + Coût de calcul raisonnable les modèles étant découplés dans la boucle interne
 - + Pas de développement du modèle adjoint couplé
 - + Contrainte faible de couplage

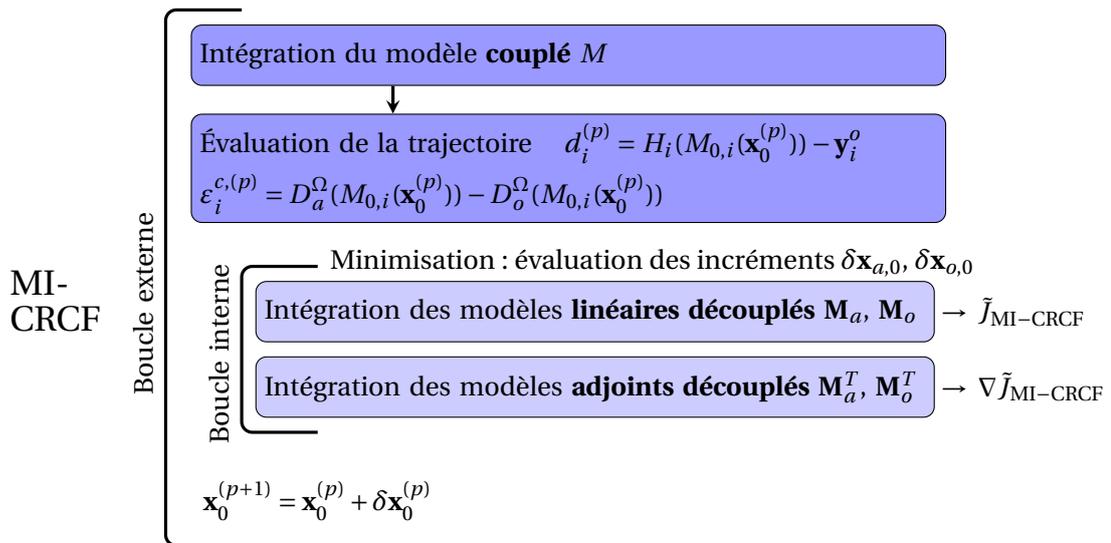


FIGURE 4.5 – Représentation schématique de l'algorithme MI-CRCF basé sur le 4D-Var incrémental. Le modèle couplé est utilisé dans la boucle externe pour évaluer la trajectoire et la minimisation est réalisée avec les deux modèles couplés indirectement via la contrainte faible de couplage J^s présente dans la fonction coût.

Cat.	Algo.	Contrôle	Modèles	Fonction Coût	±
MH	C	\mathbf{x}_0 $\Gamma : (\mathbf{x}_0, \mathbf{x}_{olr}^0)^T$ $\delta\Gamma : (\mathbf{x}_0, \delta\mathbf{x}_\Gamma)^T$	M couplé \mathbf{M} couplé \mathbf{M}^T couplé	$J = J^b + J^o$	+ Couplage - Adjoint couplé - Coût de calcul
	CR	\mathbf{x}_0 $\Gamma : (\mathbf{x}_0, \mathbf{x}_{olr}^0)^T$ $\delta\Gamma : (\mathbf{x}_0, \delta\mathbf{x}_\Gamma)^T$	M couplé \mathbf{M} couplé \mathbf{M}^T couplé	$J = J^b + J^o + \gamma J^s$	+ Couplage - Adjoint couplé ~ Coût de calcul
	CCF	\mathbf{x}_0 $\Gamma : (\mathbf{x}_0, \mathbf{x}_{olr}^0)^T$	M_α découplés \mathbf{M}_α découplés \mathbf{M}_α^T découplés	$J = J_a^b + J_o^b + J_a^o + J_o^o + \gamma J^s$	- Couplage + Adjoint découplé + Coût de calcul
MI	C	\mathbf{x}_0	M couplé \mathbf{M}_α découplés \mathbf{M}_α^T découplés	BE : $J = J^b + J^o$ BI : $\tilde{J} = J_a^b + J_a^o + J_o^b + J_o^o$	~ Couplage + Adjoint découplé + Coût de calcul
	CRCF	\mathbf{x}_0 $\Gamma : (\mathbf{x}_0, \mathbf{x}_{olr}^0)^T$ $\delta\Gamma : (\mathbf{x}_0, \delta\mathbf{x}_\Gamma)^T$	M couplé \mathbf{M}_α découplés \mathbf{M}_α^T découplés	BE : $J = J^b + J^o + J^s$ BI : $\tilde{J} = J_a^b + J_a^o + J_o^b + J_o^o + \tilde{J}^s$	~ Couplage + Adjoint découplé + Coût de calcul

Tableau 4.1 – Tableau récapitulatif des différents algorithmes d'assimilation de données couplée introduits. Les termes « BE » et « BI » permettent de désigner la fonction coût utilisée respectivement dans la boucle externe et interne du 4D-Var incrémental lorsque cette dernière n'est pas la stricte linéarisation de la première. La dernière colonne donne une indication concernant la qualité du couplage de l'analyse produite par l'algorithme, si le développement du modèle adjoint couplé est nécessaire et si la méthode sera gourmande en coût de calcul.

4.2 Étude de convergence des algorithmes d'assimilation couplée

Nous proposons dans cette section une étude de la convergence des différents algorithmes d'assimilation couplée introduits. Cette étude est basée sur les travaux de Gratton *et al.* (2007) qui étudient la convergence des algorithmes de Gauss-Newton. Nous commencerons par présenter en quoi le 4D-Var incrémental peut-être vu comme un algorithme de Gauss Newton. Nous pourrons alors en déduire des conditions suffisantes de convergence des algorithmes homogènes et inhomogènes à partir des résultats de Gratton *et al.* (2007). Enfin, nous exprimerons une condition nécessaire et suffisante de convergence des algorithmes inhomogènes dans un cadre linéaire.

4.2.1 Le 4D-Var incrémental vu comme un algorithme de Gauss-Newton

Avant toute chose, il nous semble important de présenter en quoi le 4D-Var incrémental est équivalent à l'algorithme de Gauss-Newton (noté GN dans la suite). L'algorithme GN minimise une fonction coût J telle que :

$$J(\mathbf{x}_0) = \frac{1}{2} \|\mathcal{Z}(\mathbf{x}_0)\|_2^2 \quad (4.2)$$

où \mathcal{Z} est une fonction non-linéaire vectorielle. La jacobienne \mathbf{Z} de \mathcal{Z} est alors définie telle que $\mathbf{Z}_{\mathbf{x}_0} = \left. \frac{\partial \mathcal{Z}}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0}$ et :

$$\begin{aligned}\nabla J(\mathbf{x}_0) &= \mathbf{Z}_{\mathbf{x}_0}^T \mathcal{Z}(\mathbf{x}_0) \\ \nabla^2 J(\mathbf{x}_0) &= \mathbf{Z}_{\mathbf{x}_0}^T \mathbf{Z}_{\mathbf{x}_0} + Q(\mathbf{x}_0)\end{aligned}$$

où $Q(\mathbf{x}_0)$ sont les termes de second ordre de la hessienne $\nabla^2 J(\mathbf{x}_0)$. La méthode GN est en fait une approximation de l'algorithme de Newton pour la résolution des problèmes de moindres carrés non-linéaires, qui néglige les termes d'ordre 2 de la hessienne et minimise alors J (4.2) par itérations successives telles que :

$$\begin{aligned}\delta \mathbf{x}_0^{(q)} &= -\left(\mathbf{Z}_q^T \mathbf{Z}_q\right)^{-1} \mathbf{Z}_q^T f(\mathbf{x}_0^{(q)}) \\ \mathbf{x}_0^{(q+1)} &= \mathbf{x}_0^{(q)} + \delta \mathbf{x}_0^{(q)}\end{aligned}\tag{4.3}$$

où l'on note $\mathbf{Z}_q = \mathbf{Z}_{\mathbf{x}_0^{(q)}}$ par souci de concision et $\cdot^{(q)}$ les itérations de l'algorithme. Revenons au 4D-Var incrémental et notons $G_i = H_i(M_{0,i})$ l'opérateur d'observations généralisé, \mathbf{G}_i sa linéarisation. Nous pouvons alors écrire la fonction coût linéaire de la boucle interne ainsi :

$$\begin{aligned}J(\delta \mathbf{x}_0^{(p)}) &= \frac{1}{2} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right)^T \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) + \\ &\quad \frac{1}{2} \sum_{i=0}^N \left(\mathbf{G}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)} \right)^T \mathbf{R}_i^{-1} \left(\mathbf{G}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)} \right)\end{aligned}$$

où $\cdot^{(p)}$ dénote les itérations de boucle externe. Si l'on suppose que la solution optimale est trouvée, alors nous avons que $\nabla J(\delta \mathbf{x}_0^{(p)*}) = \mathbf{0}$ (où l'exposant \cdot^* désigne la solution optimale) et :

$$\begin{aligned}\mathbf{x}_0^{(p+1)} &= \mathbf{x}_0^{(p)} + \delta \mathbf{x}_0^{(p)*} \\ &= \mathbf{x}_0^{(p)} + \left(\mathbf{B}^{-1} + \sum_{i=0}^N \mathbf{G}_i^T \mathbf{R}_i^{-1} \mathbf{G}_i \right)^{-1} \times \left(\mathbf{B}^{-1} \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} + \sum_{i=0}^N \mathbf{G}_i^T \mathbf{R}_i^{-1} d_i^{(p)} \right)\end{aligned}\tag{4.4}$$

Puisqu'il n'est pas évident que (4.3) \Leftrightarrow (4.4), définissons la fonction vectorielle \mathcal{Z} telle que :

$$\mathcal{Z}(\mathbf{x}_0) = \begin{pmatrix} \mathbf{B}^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^b) \\ \mathbf{R}_0^{-1/2}(G_0(\mathbf{x}_0) - y_0^o) \\ \dots \\ \mathbf{R}_N^{-1/2}(G_N(\mathbf{x}_0) - y_N^o) \end{pmatrix}$$

La fonction coût de la boucle externe est alors égale à $J(\mathbf{x}_0^{(p)}) = \frac{1}{2} \|\mathcal{Z}(\mathbf{x}_0^{(p)})\|_2^2$. Nous pouvons ensuite définir la jacobienne $\mathbf{Z} = \left. \frac{\partial \mathcal{Z}}{\partial \mathbf{x}_0} \right|_{\mathbf{x}_0}$ telle que :

$$\mathbf{Z} = \begin{pmatrix} \mathbf{B}^{-1/2} \\ \mathbf{R}_0^{-1/2} \mathbf{G}_0 \\ \dots \\ \mathbf{R}_N^{-1/2} \mathbf{G}_N \end{pmatrix}$$

puis montrer que :

$$-\left(\mathbf{Z}_p^T \mathbf{Z}_p\right)^{-1} \mathbf{Z}_p^T \mathcal{Z}(\mathbf{x}_0^{(p)}) = \left(\mathbf{B}^{-1} + \sum_{i=0}^N \mathbf{G}_i \mathbf{R}_i^{-1} \mathbf{G}_i \right)^{-1} \times \left(\mathbf{B}^{-1} \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} + \sum_{i=0}^N \mathbf{G}_i \mathbf{R}_i^{-1} d_i^{(p)} \right)$$

i.e. (4.3) \Leftrightarrow (4.4)

L'incrément optimal du 4D-Var incrémental (4.4) est donc égal à celui de l'algorithme GN (4.3), signifiant que les deux méthodes sont formellement équivalentes.

4.2.2 Convergence des algorithmes homogènes

Les méthodes homogènes sont simplement des algorithmes de type 4D-Var incrémental, avec l'utilisation d'un modèle couplé ou d'une fonction coût différente du 4D-Var incrémental classique. Nous venons de montrer l'équivalence entre ces méthodes et les méthodes GN, dont un critère de convergence est donné par le théorème 4 de l'article de Gratton *et al.* (2007). Nous pouvons donc en déduire une condition suffisante de convergence des méthodes homogènes :

Théorème 4.2.1 - Convergence des méthodes homogènes (d'après Gratton *et al.* (2007))¹

Soit \mathcal{Z} une fonction de H^2 , $\mathbf{Z}_{\mathbf{x}_0} = \partial_{\mathbf{x}_0} \mathcal{Z}(\mathbf{x}_0)$ sa jacobienne et $\eta \in \mathbb{R}^+$ tel que $0 \leq \eta < 1$. Alors s'il existe $\mathbf{x}_0^* \in \mathbb{R}^n$ tel que $\mathbf{Z}_{\mathbf{x}_0^*}^T \mathcal{Z}(\mathbf{x}_0^*) = 0$ avec $\mathbf{Z}_{\mathbf{x}_0^*}$ de rang n , s'il existe $\varepsilon > 0$ tel que $\|\mathbf{x}_0^b - \mathbf{x}_0^*\|_2 < \varepsilon$ et s'il existe une séquence $\eta^{(p)}$ telle que $\forall p \geq 0$:

$$\|Q(\mathbf{x}_0^{(p)}) (\mathbf{Z}_p^T \mathbf{Z}_p)^{-1}\|_2 \leq \eta^{(p)} \leq \eta$$

alors la séquence des $\mathbf{x}_0^{(p)}$ converge vers \mathbf{x}_0^* , le minimum de la fonction coût J .

Ce théorème signifie que la courbure de la fonction coût ne doit pas être trop forte pour assurer la convergence de l'algorithme. De plus, une seconde condition à la convergence est que la distance $\|\mathbf{x}_0^b - \mathbf{x}_0^*\|_2$ entre l'ébauche et l'optimum ne doit pas être trop grande. Cette distance à l'optimum est fonction de la courbure de J : ε pourra être d'autant plus grand que la courbure de J sera faible et inversement.

4.2.3 Convergence des algorithmes inhomogènes

Formalisation du problème de convergence

Boucle externe. Le problème à résoudre en boucle externe par les algorithmes inhomogènes est le suivant :

$$\begin{aligned} \text{trouver } \quad & \mathbf{x}_0^* = \underset{\mathbf{x}_0}{\operatorname{argmin}} (J(\mathbf{x}_0)) \\ \text{avec } \quad & J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N (G_i(\mathbf{x}_0) - \mathbf{y}_i^o)^T \mathbf{R}_i^{-1} (G_i(\mathbf{x}_0) - \mathbf{y}_i^o) + J^s(\mathbf{x}_0) \end{aligned} \quad (4.5)$$

où G désigne l'opérateur d'observation généralisé non-linéaire couplé. On se place dans le cas du MI-C (respectivement du MI-CRCF) en considérant $\gamma = 0$ (resp. $\gamma \neq 0$).

Boucle interne. Les MI souhaitent approcher une solution de (4.5) par itérations successives d'un problème linéaire simplifié. Soit $\delta \tilde{\mathbf{x}}_0 = \begin{pmatrix} \delta \tilde{\mathbf{x}}_{a,0} \\ \delta \tilde{\mathbf{x}}_{o,0} \end{pmatrix}$ l'incrément du vecteur de contrôle, $d_i =$

1. Pour la démonstration de ce théorème, se référer à l'article de Gratton *et al.* (2007)

$\begin{pmatrix} d_{a,i} \\ d_{o,i} \end{pmatrix}$ le vecteur d'innovations et ε_i^c l'erreur de couplage au temps t_i . La fonction coût de la boucle interne est alors la suivante :

$$\tilde{J}(\delta\tilde{\mathbf{x}}_0^{(p)}) = J_a(\delta\tilde{\mathbf{x}}_{a,0}^{(p)}) + J_o(\delta\tilde{\mathbf{x}}_{o,0}^{(p)}) + \tilde{J}^s(\delta\tilde{\mathbf{x}}_0^{(p)})$$

avec :

$$\begin{aligned} J_a(\delta\tilde{\mathbf{x}}_{a,0}^{(p)}) &= \frac{1}{2} \left(\delta\tilde{\mathbf{x}}_{a,0}^{(p)} + \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_{a,0}^{(l)} \right)^T \mathbf{B}_a^{-1} \left(\delta\tilde{\mathbf{x}}_{a,0}^{(p)} + \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_{a,0}^{(l)} \right) + \\ &\quad \frac{1}{2} \sum_{i=0}^N \left(\mathbf{G}_{a,i} \delta\tilde{\mathbf{x}}_{a,0}^{(p)} - d_{a,i}^{(p)} \right)^T \mathbf{R}_{a,i}^{-1} \left(\mathbf{G}_{a,i} \delta\tilde{\mathbf{x}}_{a,0}^{(p)} - d_{a,i}^{(p)} \right) \\ J_o(\delta\tilde{\mathbf{x}}_{o,0}^{(p)}) &= \frac{1}{2} \left(\delta\tilde{\mathbf{x}}_{o,0}^{(p)} + \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_{o,0}^{(l)} \right)^T \mathbf{B}_o^{-1} \left(\delta\tilde{\mathbf{x}}_{o,0}^{(p)} + \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_{o,0}^{(l)} \right) + \\ &\quad \frac{1}{2} \sum_{i=0}^N \left(\mathbf{G}_{o,i} \delta\tilde{\mathbf{x}}_{o,0}^{(p)} - d_{o,i}^{(p)} \right)^T \mathbf{R}_{o,i}^{-1} \left(\mathbf{G}_{o,i} \delta\tilde{\mathbf{x}}_{o,0}^{(p)} - d_{o,i}^{(p)} \right) \\ \tilde{J}^s(\delta\tilde{\mathbf{x}}_0^{(p)}) &= \frac{1}{2} \sum_{i=0}^N \left(\sqrt{\tilde{\gamma}} \mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta\tilde{\mathbf{x}}_{a,0}^{(p)} - \sqrt{\tilde{\gamma}} \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta\tilde{\mathbf{x}}_{o,0}^{(p)} + \sqrt{\tilde{\gamma}} \varepsilon_i^{c,(p)} \right)^T \mathbf{W}^{-1} \\ &\quad \left(\sqrt{\tilde{\gamma}} \mathbf{D}_a^{\Omega_a} \mathbf{M}_{a,0,i} \delta\tilde{\mathbf{x}}_{a,0}^{(p)} - \sqrt{\tilde{\gamma}} \mathbf{D}_o^{\Omega_o} \mathbf{M}_{o,0,i} \delta\tilde{\mathbf{x}}_{o,0}^{(p)} + \sqrt{\tilde{\gamma}} \varepsilon_i^{c,(p)} \right) \end{aligned}$$

où $\mathbf{G}_{\alpha,i} = \mathbf{H}_{\alpha,i} \mathbf{M}_{\alpha,0,i}$ est l'opérateur d'observation généralisé linéaire découplé ($\alpha = a, o$). Dans la suite, nous noterons les opérateurs $\tilde{\mathbf{G}}$, $\tilde{\mathbf{M}}$, \mathbf{D} et $\underline{\mathbf{D}}$ tels que :

$$\begin{aligned} \tilde{\mathbf{G}}_i \delta\tilde{\mathbf{x}}_0 &= \begin{pmatrix} \mathbf{G}_{a,i} & 0 \\ 0 & \mathbf{G}_{o,i} \end{pmatrix} \begin{pmatrix} \delta\tilde{\mathbf{x}}_{a,0} \\ \delta\tilde{\mathbf{x}}_{o,0} \end{pmatrix} & \tilde{\mathbf{M}}_{0,i} \delta\tilde{\mathbf{x}}_0 &= \begin{pmatrix} \mathbf{M}_{a,0,i} & 0 \\ 0 & \mathbf{M}_{o,0,i} \end{pmatrix} \begin{pmatrix} \delta\tilde{\mathbf{x}}_{a,0} \\ \delta\tilde{\mathbf{x}}_{o,0} \end{pmatrix} \\ \mathbf{D} &= \mathbf{D}_a^{\Omega_a} - \mathbf{D}_o^{\Omega_o} & \underline{\mathbf{D}} &= \begin{pmatrix} \mathbf{D}_a^{\Omega_a} & \\ & -\mathbf{D}_o^{\Omega_o} \end{pmatrix} \end{aligned}$$

Problème de convergence. Si l'on note \cdot^* la solution optimale d'une fonction coût, alors nous avons :

$$\begin{aligned} \tilde{\mathbf{x}}_0^{(p)*} &= \mathbf{x}^b + \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}^{(l)*} \quad \text{la solution optimale du MI-C à la boucle externe } p \\ \mathbf{x}_0^* &= \mathbf{x}^b + \delta\mathbf{x}^* \quad \text{la solution optimale du problème (4.5)} \end{aligned}$$

Le problème est de déterminer s'il existe une condition suffisante assurant la convergence des MI, et si oui, quelle sera l'erreur commise $\tilde{\varepsilon} = (\tilde{\varepsilon}_a, \tilde{\varepsilon}_o)^T$ par rapport à la solution du problème :

$$\begin{cases} \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_a^{(l)*} \xrightarrow[p \rightarrow \infty]{?} \delta\mathbf{x}_a^* + \tilde{\varepsilon}_a \\ \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_o^{(l)*} \xrightarrow[p \rightarrow \infty]{?} \delta\mathbf{x}_o^* + \tilde{\varepsilon}_o \end{cases}$$

Pour résumer, la figure 4.6 présente schématiquement le choix qui est réalisé par une MI vis à vis d'une MH. D'un côté la MH minimise la fonction coût non linéaire avec sa linéarisation exacte (utilisation du modèle couplé linéaire) en boucle interne. De l'autre la MI fait le choix de considérer deux systèmes linéaires découplés en boucle interne.

$$\begin{array}{l}
\text{Boucle externe} \left\{ \begin{array}{l}
J(\mathbf{x}_0^{(p)}) = \frac{1}{2} (\mathbf{x}_0^{(p)} - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0^{(p)} - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N (G_i(\mathbf{x}_0^{(p)}) - \mathbf{y}^o)^T \mathbf{R}^{-1} (G_i(\mathbf{x}_0^{(p)}) - \mathbf{y}^o) + \gamma J^s(\mathbf{x}_0^{(p)}) \\
\text{Boucle interne} \\
\text{MH : } J(\delta \mathbf{x}_0^{(p)}) = J^b(\delta \mathbf{x}_0^{(p)}) + \frac{1}{2} \sum_{i=0}^N (\mathbf{G}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)})^T \mathbf{R}_i^{-1} (\mathbf{G}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)}) + J^s(\delta \mathbf{x}_0^{(p)}) \\
\text{MI : } \tilde{J}(\delta \mathbf{x}_0^{(p)}) = J_a^b(\delta \mathbf{x}_{a,o}^{(p)}) + J_o^b(\delta \mathbf{x}_{o,0}^{(p)}) + \frac{1}{2} \sum_{i=0}^N (\tilde{\mathbf{G}}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)})^T \mathbf{R}_i^{-1} (\tilde{\mathbf{G}}_i \delta \mathbf{x}_0^{(p)} - d_i^{(p)}) + \tilde{J}^s(\delta \mathbf{x}_0^{(p)}) \\
\mathbf{x}_0^{(p+1)} = \mathbf{x}_0^{(p)} + \delta \mathbf{x}_0^{(p)}
\end{array} \right.
\end{array}$$

FIGURE 4.6 – Représentation schématique de la différence entre les algorithmes homogènes et inhomogènes. Les MH considèrent la linéarisation exacte de la fonction coût, alors que le MI-C en considérera une approximation via l'utilisation d'un opérateur d'observation généralisé découplé $\tilde{\mathbf{G}}$. Nous pouvons considérer $\gamma = 0$ ou $\neq 0$ au sein de J^s et \tilde{J}^s , selon le type d'algorithme choisi.

Convergence des MI vues comme des méthodes GN perturbées

Gratton *et al.* (2007) introduisent ce qu'ils appellent des méthodes de Gauss-Newton Perturbées (PGN). Il s'agit de méthodes de GN où la jacobienne de \mathcal{Z} est une approximation de la vraie jacobienne \mathbf{Z} et est notée $\tilde{\mathbf{Z}}$. Le gradient et la hessienne approchés de J sont alors écrits :

$$\begin{aligned}
\nabla \tilde{J}(\mathbf{x}_0) &= \tilde{\mathbf{Z}}_{\mathbf{x}_0}^T \mathcal{Z}(\mathbf{x}_0) \\
\nabla^2 \tilde{J}(\mathbf{x}_0) &= \tilde{\mathbf{Z}}_{\mathbf{x}_0}^T \mathbf{Z}_{\mathbf{x}_0} + \tilde{\mathbf{Q}}(\mathbf{x}_0)
\end{aligned}$$

où $\tilde{\mathbf{Q}}(\mathbf{x}_0)$ représente les termes d'ordre 2 provenant de la dérivée de $\tilde{\mathbf{Z}}_{\mathbf{x}_0}$. L'algorithme du PGN consiste à minimiser J par itérations successives en utilisant la jacobienne approchée :

$$\begin{aligned}
\delta \mathbf{x}_0^{(q)} &= -(\tilde{\mathbf{Z}}_q^T \tilde{\mathbf{Z}}_q)^{-1} \tilde{\mathbf{Z}}_q^T \mathcal{Z}(\mathbf{x}_0^{(q)}) \\
\mathbf{x}_0^{(q+1)} &= \mathbf{x}_0^{(q)} + \delta \mathbf{x}_0^{(q)}
\end{aligned} \tag{4.6}$$

avec q les itérations du PGN. Les MI peuvent alors être vues comme des méthodes PGN. En effet, il est possible de ré-écrire la fonction coût (4.5) des MI telle que :

$$\begin{aligned}
J(\mathbf{x}_0) &= \|\mathcal{Z}_{\text{MI}}(\mathbf{x}_0)\|_2^2 \\
\text{avec } \mathcal{Z}_{\text{MI}}(\mathbf{x}_0) &= \begin{pmatrix} \mathbf{B}^{-1/2}(\mathbf{x}_0 - \mathbf{x}_0^b) \\ \mathbf{R}_0^{-1/2} d_0 \\ \dots \\ \mathbf{R}_N^{-1/2} d_N \\ \sqrt{\gamma} \cdot \mathbf{W}^{-1/2} \varepsilon_0^c \\ \dots \\ \sqrt{\gamma} \cdot \mathbf{W}^{-1/2} \varepsilon_N^c \end{pmatrix}
\end{aligned}$$

où les écarts aux observations d_i et les erreurs de couplage ε_i^c sont évalués à partir du modèle couplé, tels que $d_i = G_i(\mathbf{x}_0) - \mathbf{y}_i^o = \begin{pmatrix} d_{a,i} \\ d_{o,i} \end{pmatrix}$ et $\varepsilon_i^c = (D_a^\Omega(M_{0,i}(\mathbf{x}_0)) - D_o^\Omega(M_{0,i}(\mathbf{x}_0)))$. Nous pouvons

alors écrire la jacobienne de \mathcal{Z}_{MI} et son approximation :

$$\mathbf{Z}_{\text{MI}} = \begin{pmatrix} \mathbf{B}^{-1/2} \\ \mathbf{R}_0^{-1/2} \mathbf{G}_0 \\ \dots \\ \mathbf{R}_N^{-1/2} \mathbf{G}_N \\ \sqrt{\tilde{\gamma}} \cdot \mathbf{W}^{-1/2} (\mathbf{D}\mathbf{M}_{0,0}) \\ \dots \\ \sqrt{\tilde{\gamma}} \cdot \mathbf{W}^{-1/2} (\mathbf{D}\mathbf{M}_{0,N}) \end{pmatrix} \quad \tilde{\mathbf{Z}}_{\text{MI}} = \begin{pmatrix} \mathbf{B}^{-1/2} \\ \mathbf{R}_0^{-1/2} \tilde{\mathbf{G}}_0 \\ \dots \\ \mathbf{R}_N^{-1/2} \tilde{\mathbf{G}}_N \\ \sqrt{\tilde{\gamma}} \cdot \mathbf{W}^{-1/2} (\mathbf{D}\tilde{\mathbf{M}}_{0,0}) \\ \dots \\ \sqrt{\tilde{\gamma}} \cdot \mathbf{W}^{-1/2} (\mathbf{D}\tilde{\mathbf{M}}_{0,N}) \end{pmatrix}$$

La jacobienne $\tilde{\mathbf{Z}}_{\text{MI}}$ est une approximation de \mathbf{Z}_{MI} dans le sens où l'opérateur d'observation généralisé couplé \mathbf{G}_i est approximé par l'opérateur découplé $\tilde{\mathbf{G}}_i$. De même, \mathbf{M} est approximé par l'opérateur modèle découplé $\tilde{\mathbf{M}}$ pour l'évaluation des erreurs de couplage. Enfin, il est possible de montrer que la solution optimale $\delta \tilde{\mathbf{x}}_0^{(p)*}$ de la minimisation de $\tilde{J}^{(p)}(\delta \tilde{\mathbf{x}}_0^{(p)})$ est égale à la solution optimale du PGN (4.6) :

$$\begin{aligned} \nabla \tilde{J}^{(p)}(\delta \tilde{\mathbf{x}}_0^{(p)*}) &= 0 \\ \Leftrightarrow \delta \tilde{\mathbf{x}}_0^{(p)*} &= - \left(\mathbf{B}^{-1} + \sum_{i=0}^N \tilde{\mathbf{G}}_i^T \mathbf{R}_i^{-1} \tilde{\mathbf{G}}_i + \tilde{\gamma} \sum_{i=0}^N \tilde{\mathbf{M}}_{0,i}^T \mathbf{D}^T \mathbf{W}^{-1} \mathbf{D} \tilde{\mathbf{M}}_{0,i} \right)^{-1} \times \\ &\quad \left(\mathbf{B}^{-1} \sum_{l=0}^{p-1} \delta \tilde{\mathbf{x}}_0^{(l)*} - \sum_{i=0}^N \tilde{\mathbf{G}}_i^T \mathbf{R}_i^{-1} d_i^{(p)} + \sqrt{\tilde{\gamma}} \sqrt{\tilde{\gamma}} \sum_{i=0}^N \tilde{\mathbf{M}}_{0,i}^T \mathbf{D}^T \mathbf{W}^{-1} \varepsilon_i^{c(p)} \right) \\ &= - \left(\tilde{\mathbf{Z}}_{p,\text{MI}}^T \tilde{\mathbf{Z}}_{p,\text{MI}} \right)^{-1} \tilde{\mathbf{Z}}_{p,\text{MI}}^T \mathcal{Z}_{\text{MI}}(\mathbf{x}_0^{(p)}) \end{aligned}$$

où $\tilde{\mathbf{Z}}_{p,\text{MI}}$ désigne la jacobienne approchée de \mathcal{Z}_{MI} à l'itération p . Les méthodes inhomogènes sont donc formellement équivalentes à des algorithmes PGN. Il est alors possible d'en déterminer un critère de convergence suffisant à partir du théorème 6 de Gratton *et al.* (2007) qui prouve la convergence des algorithmes PGN sous certaines conditions :

Théorème 4.2.2 - Convergence des méthodes inhomogènes (d'après Gratton *et al.* (2007))²

Soit \mathcal{Z} une fonction de H^2 , $\mathbf{Z}_{\mathbf{x}_0} = \partial_{\mathbf{x}_0} \mathcal{Z}(\mathbf{x}_0)$ sa jacobienne et $\eta \in \mathbb{R}^+$ tel que $0 \leq \eta < 1$. Alors s'il existe $\tilde{\mathbf{x}}_0^* \in \mathbb{R}^n$ tel que $\tilde{\mathbf{Z}}_{\tilde{\mathbf{x}}_0^*}^T \mathcal{Z}(\tilde{\mathbf{x}}_0^*) = 0$ avec $\tilde{\mathbf{Z}}_{\tilde{\mathbf{x}}_0^*}$ de rang n , s'il existe $\varepsilon > 0$ tel que $\|\mathbf{x}_0^b - \tilde{\mathbf{x}}_0^*\|_2 \leq \varepsilon$ et s'il existe une séquence $\eta^{(p)}$ telle que $\forall p \geq 0$:

$$\|\mathbf{I} - \left(\tilde{\mathbf{Z}}_p^T \mathbf{Z}_p + \tilde{Q}(\mathbf{x}_0^{(p)}) \right) \left(\tilde{\mathbf{Z}}_p^T \tilde{\mathbf{Z}}_p \right)^{-1}\|_2 \leq \eta^{(p)} \leq \eta$$

alors la séquence des $\mathbf{x}_0^{(p)}$ converge vers $\tilde{\mathbf{x}}_0^*$.

Ce théorème nous dit que pour assurer la convergence des MI, il est nécessaire que $\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}}$ soit une bonne approximation de $\nabla^2 \tilde{J}(\mathbf{x}_0)$. De plus, on remarquera que bien que l'algorithme ait convergé, rien ne nous assure qu'il ait convergé vers la solution de la boucle externe. En effet, nous avons à l'optimal $\nabla \tilde{J}(\tilde{\mathbf{x}}_0^*) = 0$, mais pas nécessairement $\nabla J(\tilde{\mathbf{x}}_0^*) = 0$. Le théorème 7 de l'article de Gratton *et al.* (2007) nous donne une majoration de l'erreur commise :

$$\|\tilde{\mathbf{x}}_0^* - \mathbf{x}_0^*\|_2 \leq \frac{1}{1 - \nu} \left\| \left(\tilde{\mathbf{Z}}_{\tilde{\mathbf{x}}_0^*}^+ - \mathbf{Z}_{\tilde{\mathbf{x}}_0^*}^+ \right) f(\tilde{\mathbf{x}}_0^*) \right\|_2 \quad (4.7)$$

2. Pour la démonstration de ce théorème, se référer à l'article de Gratton *et al.* (2007)

où $\mathbf{Z}^+ = (\mathbf{Z}^T \mathbf{Z})^{-1} \mathbf{Z}^T$ est appelé inverse de Moore-Penrose et $\nu < 1$ dépend des termes de second ordre $Q(\mathbf{x}_0)$. La convergence est exacte ($\tilde{\mathbf{x}}_0^* = \mathbf{x}_0^*$) si $J(\mathbf{x}_0^*) = 0^3$.

Remarque 4.2.1

Si nous remplaçons dans le théorème 4.2.2 la jacobienne approchée $\tilde{\mathbf{J}}$ par la jacobienne exacte \mathbf{J} , alors nous retrouvons le critère de convergence du GN (théorème 4.2.1).

En pratique, il peut être compliqué de déterminer si cette condition est respectée. En effet, il n'est pas toujours aisé d'exprimer \tilde{Q}_{MI} . De plus, ce critère est une condition suffisante et non nécessaire, il est alors possible qu'il ne soit pas atteint et que l'algorithme converge tout de même. Nous proposons donc de réaliser l'étude de convergence des méthodes MI dans un cadre linéaire, afin d'exprimer un facteur de convergence nécessaire et suffisant.

Convergence des MI dans le cas linéaire

On considère la fonction coût quadratique suivante à minimiser :

$$J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{x}_0 - \mathbf{x}_0^b)^T \mathbf{B}^{-1} (\mathbf{x}_0 - \mathbf{x}_0^b) + \frac{1}{2} \sum_{i=0}^N (\mathbf{G}_i \mathbf{x}_0 - \mathbf{y}_i^o)^T \mathbf{R}_i^{-1} (\mathbf{G}_i \mathbf{x}_0 - \mathbf{y}_i^o) + \gamma \frac{1}{2} \sum_{i=0}^N (\mathbf{D}\mathbf{M}_{0,i} \mathbf{x}_0)^T \mathbf{W}^{-1} (\mathbf{D}\mathbf{M}_{0,i} \mathbf{x}_0) \quad (4.8)$$

Nous allons simplifier légèrement le problème, sans en perdre sa généralité, en considérant les observations à un unique pas de temps. Nous pouvons alors écrire que :

$$J(\mathbf{x}_0) = \frac{1}{2} (\mathbf{\Pi} \mathbf{x}_0 - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{\Pi} \mathbf{x}_0 - \mathbf{y})$$

avec $\mathbf{\Pi} = \begin{pmatrix} \mathbf{I} \\ \mathbf{G} \\ \sqrt{\gamma} \mathbf{D}\mathbf{M}_{0,0} \\ \vdots \\ \sqrt{\gamma} \mathbf{D}\mathbf{M}_{0,N} \end{pmatrix}$, $\mathbf{y} = \begin{pmatrix} \mathbf{x}_0^b \\ \mathbf{y}^o \\ 0 \\ \vdots \\ 0 \end{pmatrix}$ et $\mathbf{S} = \begin{pmatrix} \mathbf{B} & 0 & 0 \\ 0 & \mathbf{R} & 0 \\ 0 & 0 & \mathbf{W} \end{pmatrix}$

La fonction coût simplifiée minimisée en boucle interne est alors la suivante :

$$\tilde{J}(\delta \tilde{\mathbf{x}}_0^{(p)}) = \frac{1}{2} (\tilde{\mathbf{\Pi}} \delta \tilde{\mathbf{x}}_0^{(p)} - \mathbf{d}^{(p)})^T \mathbf{S}^{-1} (\tilde{\mathbf{\Pi}} \delta \tilde{\mathbf{x}}_0^{(p)} - \mathbf{d}^{(p)})$$

avec $\tilde{\mathbf{\Pi}} = \begin{pmatrix} \mathbf{I} \\ \tilde{\mathbf{G}} \\ \sqrt{\tilde{\gamma}} \tilde{\mathbf{D}}\tilde{\mathbf{M}}_{0,0} \\ \vdots \\ \sqrt{\tilde{\gamma}} \tilde{\mathbf{D}}\tilde{\mathbf{M}}_{0,N} \end{pmatrix}$ et $\mathbf{d}^{(p)} = \mathbf{y} - \mathbf{\Pi} \mathbf{x}_0^{(p)} = \begin{pmatrix} -\sum_{l=0}^{p-1} \delta \tilde{\mathbf{x}}_0^{(l)} \\ \mathbf{d}^{(p)} \\ -\sqrt{\gamma} \varepsilon_0^{c,(p)} \\ \vdots \\ -\sqrt{\gamma} \varepsilon_N^{c,(p)} \end{pmatrix}$

Notons que le cas $\tilde{\mathbf{\Pi}} = \mathbf{\Pi}$ correspond à un algorithme homogène avec un modèle linéaire. La solution optimale qui minimise $\tilde{J}(\delta \tilde{\mathbf{x}}_0^{(p)})$ à l'itération p est donnée par la solution de $\nabla \tilde{J}(\delta \tilde{\mathbf{x}}_0^{(p)}) = \mathbf{0}$:

$$\delta \tilde{\mathbf{x}}_0^{(p)*} = \tilde{\mathbf{K}} \mathbf{d}^{(p)}$$

avec $\tilde{\mathbf{K}} = (\tilde{\mathbf{\Pi}}^T \mathbf{S}^{-1} \tilde{\mathbf{\Pi}})^{-1} \tilde{\mathbf{\Pi}}^T \mathbf{S}^{-1}$

3. Pour retrouver la majoration de l'erreur commise (4.7), se référer à l'article Gratton *et al.* (2007)

Les opérateurs étant tous linéaires, il est possible de définir $\mathbf{d}^{(p)}$ en fonction de $\delta\tilde{\mathbf{x}}^{(p-1)}$:

$$\begin{aligned}\mathbf{d}^{(p)} &= \mathbf{y} - \Pi\mathbf{x}_0^{(p)} \\ &= \mathbf{y} - \Pi\mathbf{x}_0^{(p-1)} - \Pi\delta\tilde{\mathbf{x}}_0^{(p-1)} \\ &= \mathbf{d}^{(p-1)} - \Pi\delta\tilde{\mathbf{x}}_0^{(p-1)}\end{aligned}$$

Nous pouvons alors écrire $\delta\tilde{\mathbf{x}}_0^{(p)*}$ en fonction de $\delta\tilde{\mathbf{x}}^{(p-1)}$, $\forall p > 1$:

$$\begin{aligned}\delta\tilde{\mathbf{x}}_0^{(p)*} &= \tilde{\mathbf{K}}\mathbf{d}^{(p)} \\ &= \tilde{\mathbf{K}}\left(\mathbf{d}^{(p-1)} - \Pi\delta\tilde{\mathbf{x}}_0^{(p-1)*}\right) \\ &= \delta\tilde{\mathbf{x}}_0^{(p-1)*} - \tilde{\mathbf{K}}\Pi\delta\tilde{\mathbf{x}}_0^{(p-1)*} \\ &= (\mathbf{I} - \tilde{\mathbf{K}}\Pi)\delta\tilde{\mathbf{x}}_0^{(p-1)*}\end{aligned}$$

$\delta\tilde{\mathbf{x}}_0^{(p)*}$ s'exprime donc comme une suite géométrique de raison $(\mathbf{I} - \tilde{\mathbf{K}}\Pi)$. Nous pouvons alors en déduire un critère de convergence nécessaire et suffisant des algorithmes inhomogènes dans le cadre linéaire (théorème 4.2.3).

Théorème 4.2.3 - Convergence des MI dans un cadre linéaire

Considérons la fonction coût J à minimiser linéaire (4.8) ($J \in H^1$). Un algorithme inhomogène converge si et seulement si :

$$\|\mathbf{I} - \tilde{\mathbf{K}}\Pi\|_2 < 1$$

Si la condition de convergence du théorème 4.2.3 est respectée, alors nous pouvons exprimer l'incrément optimal total du MI. En effet, nous avons que :

$$\begin{aligned}\delta\tilde{\mathbf{x}}_0^{(p)*} &= (\mathbf{I} - \tilde{\mathbf{K}}\Pi)^p \delta\tilde{\mathbf{x}}_0^{(0)*} \\ \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_0^{(l)*} &= \sum_{l=0}^{p-1} (\mathbf{I} - \tilde{\mathbf{K}}\Pi)^l \delta\tilde{\mathbf{x}}_0^{(1)*} \\ \text{donc : } \sum_{l=0}^{p-1} \delta\tilde{\mathbf{x}}_0^{(l)*} &\xrightarrow{p \rightarrow \infty} \sum_{l=0}^{\infty} (\mathbf{I} - \tilde{\mathbf{K}}\Pi)^l \tilde{\mathbf{K}}\mathbf{d}^{(0)}\end{aligned}\tag{4.9}$$

On remarque que la somme vers laquelle converge l'incrément total du MI (équation (4.9)) est une série de Neumann :

Définition 4.2.1 - Série de Neumann

Une série de Neumann est une série de la forme :

$$\sum_{l=0}^{\infty} \mathbf{T}^l$$

où \mathbf{T} est un opérateur linéaire ici. Si $\|\mathbf{T}\|_2 < 1$, alors $(\mathbf{I} - \mathbf{T})$ est inversible et :

$$(\mathbf{I} - \mathbf{T})^{-1} = \sum_{l=0}^{\infty} \mathbf{T}^l$$

Ainsi, si la MI converge, alors nous avons que :

$$\sum_{l=0}^{p-1} \delta \tilde{\mathbf{x}}_0^{(l)*} \xrightarrow{p \rightarrow \infty} (\tilde{\mathbf{K}}\mathbf{\Pi})^{-1} \tilde{\mathbf{K}}\mathbf{d}^{(0)} \quad (4.10)$$

et nous pouvons exprimer l'erreur $\tilde{\varepsilon}$ commise par la MI par rapport à la solution optimale du problème \mathbf{x}_0^* . En effet $\tilde{\varepsilon} = \mathbf{x}_0^* - \tilde{\mathbf{x}}_0^*$, où $\mathbf{x}_0^* = \mathbf{x}_0^b + \delta \mathbf{x}_0^*$ avec :

$$\delta \mathbf{x}_0^* = \mathbf{K}\mathbf{d}^{(0)} \quad (4.11)$$

où $\mathbf{K} = (\mathbf{\Pi}^T \mathbf{S}^{-1} \mathbf{\Pi})^{-1} \mathbf{\Pi}^T \mathbf{S}^{-1}$. Le théorème 4.2.4 définit alors l'erreur commise par la MI à partir des équations (4.10) et (4.11).

Théorème 4.2.4 - Erreur des MI dans un cadre linéaire

Supposons la fonction coût J à minimiser linéaire (4.8) et que la MI ait convergé (théorème 4.2.3 vérifié), alors nous pouvons exprimer l'erreur $\tilde{\varepsilon}$ commise par rapport à la solution optimale de J :

$$\tilde{\varepsilon} = \left(\mathbf{K} - (\tilde{\mathbf{K}}\mathbf{\Pi})^{-1} \tilde{\mathbf{K}} \right) \mathbf{d}^{(0)}$$

Nous pouvons faire plusieurs remarques sur ce résultat de convergence des MI dans un cadre linéaire.

Remarque 4.2.2 - Cohérence du critère de convergence

Nous pouvons remarquer que les MH dans le cas linéaire convergeront en une seule itération de boucle externe, ce qui est cohérent. Pour s'en persuader, considérons l'expression de l'incrément optimal (4.9) : $\delta \tilde{\mathbf{x}}_0^{(p)*} = (\mathbf{I} - \tilde{\mathbf{K}}\mathbf{\Pi})^p \delta \tilde{\mathbf{x}}_0^{(0)*}$. Dans le cadre des MH, nous aurons \mathbf{K} à la place de $\tilde{\mathbf{K}}$, et alors $\mathbf{K}\mathbf{\Pi} = \mathbf{I}$ impliquant que $\forall p > 1 : \delta \tilde{\mathbf{x}}_0^{(p)*} = \mathbf{0}$:

$$\begin{aligned} \mathbf{K}\mathbf{\Pi} &= (\mathbf{\Pi}^T \mathbf{S}^{-1} \mathbf{\Pi})^{-1} \mathbf{\Pi}^T \mathbf{S}^{-1} \mathbf{\Pi} \\ &= \mathbf{I} \end{aligned}$$

Remarque 4.2.3 - Lien avec le critère de convergence des PGN

Gratton *et al.* (2007) ont exprimé le critère de convergence suffisant des méthodes PGN dans un cadre non linéaire suivant :

$$\forall p \geq 0 \quad \left\| \mathbf{I} - \left(\tilde{\mathbf{Z}}_p^T \mathbf{Z}_p + \tilde{Q}(\mathbf{x}_0^{(p)}) \right) \left(\tilde{\mathbf{Z}}_p^T \tilde{\mathbf{Z}}_p \right)^{-1} \right\|_2 < 1$$

Dans un cadre linéaire, ce critère devient :

$$\left\| \mathbf{I} - (\tilde{\mathbf{Z}}^T \mathbf{Z}) (\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}})^{-1} \right\|_2 < 1$$

Il ne s'agit cependant que d'une condition suffisante de convergence. Nous avons pu ici exprimer une condition nécessaire et suffisante de convergence dans le théorème 4.2.3, que nous pouvons ré-écrire en fonction des matrices jacobienne ainsi :

$$\left\| \mathbf{I} - (\tilde{\mathbf{Z}}^T \tilde{\mathbf{Z}})^{-1} (\tilde{\mathbf{Z}}^T \mathbf{Z}) \right\|_2 < 1$$

Nous pourrions évaluer numériquement dans la suite la différence entre ces deux critères.

Remarque 4.2.4 - Impact de la contrainte faible de couplage

La présence ou non de la contrainte faible de couplage au sein de l'algorithme aura un impact sur la convergence des MI. En effet, nous pouvons distinguer plusieurs cas :

- $(\gamma = 0, \tilde{\gamma} = 0)$: il s'agit de l'algorithme MI-C. La convergence sera uniquement conditionnée par « l'écart » entre les modèles de la boucle externe et interne. Si \mathbf{G} est « proche » de $\tilde{\mathbf{G}}$, alors il y a de grandes chances pour que $\|\mathbf{I} - \tilde{\mathbf{K}}\mathbf{\Pi}\|_2 < 1$ et que l'algorithme converge. Le MI-C convergera donc si la suppression du couplage entre les modèles n'est pas une trop forte approximation.
- $(\gamma \neq 0, \tilde{\gamma} \neq 0)$: dans ce cas, nous cherchons à minimiser les erreurs d'interface dans la boucle externe et interne. Nous pouvons espérer que $\tilde{\gamma} \neq 0$ permettra de compenser l'erreur commise dans l'approximation de \mathbf{G} . Cependant, la contrainte supplémentaire imposée par $\gamma \neq 0$ peut limiter cette compensation.
- $(\gamma = 0, \tilde{\gamma} \neq 0)$: ce dernier cas est peut-être le meilleur candidat pour satisfaire la condition de convergence. En effet, la contrainte faible n'est présente que dans la boucle interne permettant *a priori* de mieux compenser les erreurs d'approximation du modèle.

Pour conclure, il a pu être mis en évidence un critère de convergence nécessaire et suffisant permettant la convergence des MI dans un cadre linéaire. Notons que ce critère ne sera pas nécessairement simple à atteindre, surtout dans le cadre des méthodes type MI-C, où supprimer le couplage entre les modèles pourrait s'avérer être une trop forte approximation. Nous pouvons alors dans ce cas espérer que l'introduction d'un terme de pénalisation J^s permettra de compenser ces erreurs d'approximation pour satisfaire le critère de convergence.

Chapitre 5

Évaluation numérique de quelques algorithmes d'assimilation couplée dans un cadre linéaire

Sommaire

5.1 Description d'un modèle simple colonne de diffusion	85
5.1.1 Équations du modèle couplé	85
5.1.2 Cas test de référence	87
5.2 Assimilation d'une observation colonne unique	88
5.2.1 Ébauche, observations et contrainte faible	88
5.2.2 Description des expériences jumelles	89
5.2.3 Résultats	92
5.2.4 Convergence des méthodes inhomogènes	95
5.3 Discussions et perspectives	99

Nous venons dans le chapitre précédent d'introduire un certain nombre d'algorithmes visant à assimiler des données au sein d'un système couplé. Nous allons dans ce chapitre effectuer quelques premiers tests numériques sur ces algorithmes. Nous allons pour cela considérer un modèle couplé de diffusion à une dimension sur lequel nous tenterons d'assimiler des données de manière couplée. Nous choisissons d'utiliser un modèle très simple dans le but de mieux comprendre comment se comportent les algorithmes d'assimilation. Cela nous permettra également de vérifier numériquement les résultats obtenus précédemment sur la convergence des différentes méthodes.

5.1 Description d'un modèle simple colonne de diffusion

5.1.1 Équations du modèle couplé

Commençons par décrire les équations régissant notre modèle 1D vertical de diffusion. Soit $\Omega \subset \mathbb{R}$ le domaine spatial de définition de notre modèle couplé. Nous décomposons Ω en

deux sous-domaines $\Omega_a = [0, h_a]$ et $\Omega_o[-h_o, 0]$ partageant une interface commune $\Gamma = 0$. Nous considérerons le modèle couplé suivant, intégré sur la fenêtre temporelle $[0, T]$:

$$\begin{cases} \mathcal{M}_o(x_o^k) = f_o & \text{sur } \Omega_o \times [0, T] \\ x_o^k(0, z) = x_{o,0} & \text{avec } z \in \Omega_o \\ \mathcal{D}_o(x_{o|\Gamma}^k) = \mathcal{D}_{ao}(x_{a|\Gamma}^{k-1}) & \text{sur } \Gamma \times [0, T] \end{cases} \quad \begin{cases} \mathcal{M}_a(x_a^k) = f_a & \text{sur } \Omega_a \times [0, T] \\ x_a^k(0, z) = x_{a,0} & \text{avec } z \in \Omega_a \\ \mathcal{D}_a(x_{a|\Gamma}^k) = \mathcal{D}_{oa}(x_{o|\Gamma}^k) & \text{sur } \Gamma \times [0, T] \end{cases} \quad (5.1)$$

où \mathcal{M}_α sont deux opérateurs de diffusion tels que :

$$\mathcal{M}_\alpha = \frac{\partial}{\partial t} + v_\alpha \frac{\partial^2}{\partial z^2}$$

avec $v_a \neq v_o$. Les opérateurs d'interfaces sont choisis comme des opérateurs de Dirichlet et de Neumann :

$$\begin{cases} \mathcal{D}_o = \text{Id} \\ \mathcal{D}_{ao} = \text{Id} \end{cases} \quad \begin{cases} \mathcal{D}_a = v_a \frac{\partial}{\partial z} \\ \mathcal{D}_{oa} = v_o \frac{\partial}{\partial z} \end{cases}$$

Il s'agit d'un modèle couplé par algorithme de Schwarz mais nous pourrions également considérer le cas d'un couplage *one-way* en stoppant l'algorithme après la première itération ou encore deux modèles découplés si nous supprimons le couplage. Le couplage *one-way* est en fait équivalent au couplage asynchrone introduit précédemment, restreint à une seule fenêtre temporelle. Dans le cas de deux modèles découplés, nous avons alors besoin de deux conditions initiales à l'interface $x_{a|\Gamma}^0$ et $x_{o|\Gamma}^0$. Notons $x = (x_a, x_o)^T$ le vecteur d'état du système et $x_0 = (x_{a,0}, x_{o,0})^T$ l'état initial à $t = 0$, alors nous désignerons dans la suite :

- $x^{\text{cvg}}(t) = \mathcal{M}_{0,t}^{\text{cvg}}(x_0, x_{a|\Gamma}^0)$ la solution du modèle couplé à convergence de Schwarz,
- $x^{k_{\text{max}}}(t) = \mathcal{M}_{0,t}^{k_{\text{max}}}(x_0, x_{a|\Gamma}^0)$ la solution du modèle couplé stoppé à l'itération k_{max} ,
- $x^{\text{unc}}(t) = \mathcal{M}_{0,t}^{\text{unc}}(x_0, x_{a|\Gamma}^0, x_{o|\Gamma}^0)$ la solution du modèle découplé.

Nous définissons le critère de convergence de l'algorithme de Schwarz ε tel que pour une itération k :

$$\varepsilon = \|\mathcal{D}_o(x_{o|\Gamma}^k) - \mathcal{D}_{ao}(x_{a|\Gamma}^k)\|_2$$

Pour compléter la description du modèle, il nous faut décrire les seconds membres des équations du modèle (5.1) qui seront fonction de la solution analytique du modèle.

Solution analytique

Nous choisissons la solution analytique du modèle suivante sur chacun des deux sous-domaines :

$$x_\alpha^*(z, t) = \frac{X_0}{4} e^{-\frac{|z|}{\beta_\alpha}} \left(3 + \cos^2 \left(\frac{3\pi t}{\tau} \right) \right) \quad \text{sur } \Omega_\alpha \times [0, T] \quad (5.2)$$

où X_0 correspond à l'amplitude de la solution. Notons que pour assurer la continuité du flux à l'interface Γ , il nous faut considérer que $\beta_a v_o = \beta_o v_a$. Nous choisirons alors les constantes suivantes du problème :

$$\begin{array}{lll} X_0 = 20 \text{ }^\circ\text{C} & v_a = 1 \text{ m}^2 \cdot \text{s}^{-1} & v_o = 0.1 \text{ m}^2 \cdot \text{s}^{-1} \\ \tau = 22 \text{ h} & \beta_a = 4000 \text{ m} & \beta_o = 400 \text{ m} \end{array}$$

Les seconds membres des équations f_α sont ensuite déterminés tels que $\mathcal{M}_\alpha(x_\alpha^*) = f_\alpha$.

5.1.2 Cas test de référence

Maintenant que le modèle est entièrement défini, il nous faut le discrétiser puis définir un état de référence et une ébauche à partir desquelles nous pourrions assimiler des données pour tester nos algorithmes.

Discrétisation

Nous discrétisons le modèle (5.1) avec un schéma d'Euler implicite en temps, et du second ordre en espace. Les résolutions seront les suivantes :

$$\begin{aligned} \Delta t = \Delta t_a = \Delta t_o = 180 \text{ s} & \quad \Delta z = \Delta z_a = \Delta z_o = 20 \text{ m} \\ T = 12 \text{ h} & \quad h_a = h_o = 1000 \text{ m} \end{aligned}$$

La fenêtre temporelle $[0, T]$ est ainsi décomposée en $N_T + 1 = 241$ pas de temps t_i avec $i \in \llbracket 0, N_T \rrbracket$ et $t_i = i \cdot \Delta t$. Les domaines Ω_α sont quant à eux décomposés en 51 niveaux verticaux chacun.

État vrai

Nous définissons l'état vrai sur la fenêtre $[0, T]$ comme étant la solution à convergence du modèle (5.1) dont l'état initial est défini à partir de la solution analytique (5.2) :

$$\mathbf{x}_0^t = \begin{pmatrix} \mathbf{x}_{a,0}^t \\ \mathbf{x}_{o,0}^t \end{pmatrix} = \begin{pmatrix} \mathbf{x}_{a,0}^* \\ \mathbf{x}_{o,0}^* \end{pmatrix}$$

où $\mathbf{x}_{\alpha,0}^*$ est la discrétisation de $x_\alpha^*(z, 0)$ (5.2) avec $z \in \Omega_\alpha$. Nous choisissons comme first-guess à l'interface $\mathbf{x}_{a,\Gamma}^0$ un bruit blanc gaussien. Le critère de convergence choisi pour les itérations de Schwarz est $\varepsilon = 10^{-6}$. Nous obtenons alors l'état vrai $\mathbf{x}_i^t \in \Omega \forall i \in \llbracket 0, N_T \rrbracket$ décrit par la figure 5.1.

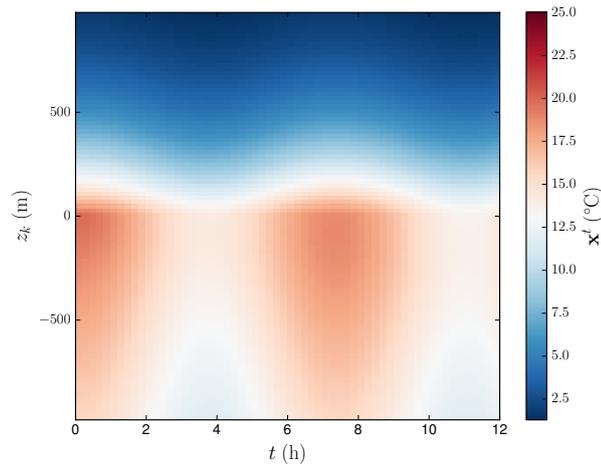


FIGURE 5.1 – État vrai considéré pour le modèle couplé de diffusion 1D.

Nous allons pouvoir maintenant tenter d'assimiler des données avec des observations prises au dernier pas de temps ($t = 12 \text{ h}$) sur tout le domaine Ω .

5.2 Assimilation d'une observation colonne unique

Nous utiliserons pour l'ensemble de nos expériences d'assimilation un algorithme du gradient conjugué préconditionné (PCG) pour minimiser la fonction coût dans la boucle interne du 4D-Var incrémental. On définit la convergence de cet algorithme à partir de son résidu $r^{(q)}$ à chaque itération de minimisation q , tel que $r^{(q)} = b - \mathbf{Ax}^{(q)}$, avec $\mathbf{Ax} = b$ le système à résoudre. Nous définissons alors la condition de convergence du PCG suivant :

$$\frac{\|r^{(q)}\|}{\|r^{(0)}\|} < \varepsilon_{\text{PCG}} \quad (5.3)$$

où ε_{PCG} est le critère de convergence de l'algorithme. Dans la suite nous considérerons $\varepsilon_{\text{PCG}} = 10^{-10}$. Nous pouvons trouver une description complète de cet algorithme dans le livre de Quarteroni *et al.* (2007) par exemple.

5.2.1 Ébauche, observations et contrainte faible

Afin de comparer les différents algorithmes, nous allons réaliser diverses expériences jumelles, c'est à dire à partir de la même ébauche, du même jeu d'observations et des mêmes matrices de covariance d'erreur. Cela nous permettra de pouvoir discriminer au mieux les méthodes. Nous proposons de décrire ici l'ébauche, les observations et la contrainte faible J^s présente dans certaines des méthodes d'assimilation couplée introduites dans le chapitre précédent.

Ébauche

L'ébauche est fournie par la solution du modèle à convergence de Schwarz à partir d'un état initial biaisé par rapport à celui du true-state. Le biais est appliqué à l'amplitude de la condition initiale, X_0 (équation (5.2)), et vaut $\delta X_0 = 5^\circ\text{C}$. La figure 5.2 représente cette ébauche \mathbf{x}^b . Nous y associons une matrice de covariance \mathbf{B} diagonale telle que $\mathbf{B} = 100 \cdot \mathbf{I}$.

Remarque 5.2.1

La matrice de covariance d'erreur d'ébauche n'a ici pas de rapport avec le biais appliqué. Nous la considérons simplement comme un poids appliqué à l'ébauche, sans lien direct avec son erreur par rapport à l'état vrai.

Observations

Nous considérons des observations parfaites, prises sur toute la colonne du dernier pas de temps de l'état vrai, soit : $\mathbf{y}^o = \mathbf{x}_{N_t}^t$ ($H = Id$). Des observations parfaites impliquent une matrice de covariance d'erreur \mathbf{R} nulle. Cependant, pour notre étude, nous associons à ces observations une matrice de covariance d'erreur \mathbf{R} diagonale telle que : $\mathbf{R} = 10 \cdot \mathbf{I}$. De même que pour l'ébauche, cette matrice est uniquement vu comme un poids, et n'a pas de lien direct avec les erreurs d'observations.

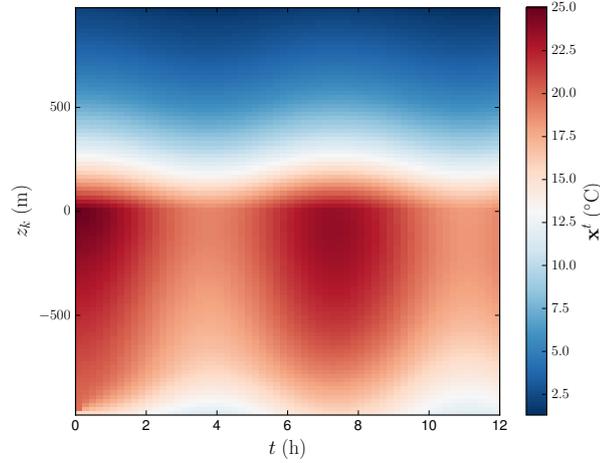


FIGURE 5.2 – Ébauche considérée pour le modèle couplé de diffusion 1D.

Contrainte faible de couplage

La contrainte faible de couplage J^s présente dans la fonction coût de certains des algorithmes quantifie l'erreur de couplage commise à l'interface :

$$J^s(\mathbf{x}_0) = \frac{1}{2}\gamma \left[\sum_{i=0}^{N_T} (D_o^\Omega(M_i(\mathbf{x}_0)) - D_{ao}^\Omega(M_i(\mathbf{x}_0)))^T \mathbf{W}_D^{-1} (D_o^\Omega(M_i(\mathbf{x}_0)) - D_{ao}^\Omega(M_i(\mathbf{x}_0))) + \sum_{i=0}^{N_T} (D_a^\Omega(M_i(\mathbf{x}_0)) - D_{oa}^\Omega(M_i(\mathbf{x}_0)))^T \mathbf{W}_N^{-1} (D_a^\Omega(M_i(\mathbf{x}_0)) - D_{oa}^\Omega(M_i(\mathbf{x}_0))) \right] \quad (5.4)$$

avec D_α^Ω les extensions des opérateurs D_α sur le domaine Ω , comme cela a été défini dans le chapitre précédent. Les matrices de covariance \mathbf{W}_D et \mathbf{W}_N sont définies telles que :

$$\mathbf{W}_D^{-1} = \mathbf{w}_D \cdot I \quad \mathbf{W}_N^{-1} = \mathbf{w}_N \cdot I$$

où \mathbf{w}_D et \mathbf{w}_N sont les poids relatifs aux erreurs de couplage de type Dirichlet et de type Neumann. Or, ces deux types d'erreur représentent des flux de nature différente. Il nous faudra donc considérer le rapport suivant entre les deux poids pour que ces termes aient une influence équivalente sur le poids total de la fonction coût, et soient alors comparables :

$$\mathbf{w}_N = \frac{\beta_a}{\nu_a} \mathbf{w}_D$$

et nous choisirons dans la suite $\mathbf{w}_D = 1$ et $\tilde{\gamma} = \gamma$. Nous jouerons alors dans la suite sur le coefficient γ pour évaluer l'impact de la contrainte faible de couplage.

5.2.2 Description des expériences jumelles

Nous allons réaliser diverses expériences jumelles que nous décrirons au fur et à mesure de notre réflexion. Le but de ces expériences sera de mieux comprendre différents aspects de l'assimilation couplée. Nous pourrions par exemple évaluer l'impact sur l'analyse d'un modèle fortement couplé par rapport à un modèle faiblement couplé, mais également l'apport de la contrainte faible de couplage J^s . Nous observerons également les performances des méthodes inhomogènes vis à vis des méthodes homogènes. Afin de comparer les différents algorithmes nous utiliserons, entre autres, deux indicateurs :

- l'erreur quadratique moyenne RMSE : $\sqrt{\frac{1}{N_T} \sum_{i=0}^{N_T} (\mathbf{x}_i^a - \mathbf{x}_i^t)^2}$ sur $\Omega \times [0, T]$, où \mathbf{x}^a dénote l'analyse fournie par un algorithme d'assimilation,
- l'erreur de couplage à l'interface (ECI), égale à $\gamma J^S(\mathbf{x}_0^a)$ avec $\gamma = 0.01$. L'ECI permet d'évaluer la qualité du couplage de l'analyse. Plus l'ECI est proche de 0, plus le couplage sera fort mathématiquement.

Nomenclature

Puisque nous allons réaliser de nombreuses expériences, il nous semble important d'introduire une nomenclature permettant rapidement de catégoriser une expérience à partir de son acronyme. Une expérience sera désignée par :

- le nom de l'algorithme considéré sous forme d'acronyme. Ces différents acronymes ont été introduits dans le chapitre précédent lors de l'introduction des différentes méthodes d'assimilation de données couplée.
- une lettre ou un chiffre permettant de désigner la force du couplage. Dans certains cas cela indiquera le nombre d'itérations de Schwarz, dans d'autres le poids relatif à J^S .

Remarque 5.2.2 - Rappel concernant la nomenclature des méthodes introduites

Afin de mieux comprendre la nomenclature qui sera utilisé, nous rappelons rapidement celle que nous avons introduite dans le chapitre précédent. Tout d'abord, nous distinguons les méthodes homogènes (MH) des méthodes inhomogènes (MI). Ensuite, chaque méthode présente différentes variantes, par exemple la variante Couplée (MH-C et MI-C), considérant une fonction coût classique avec un modèle couplé. Puis les MH Couplées Renforcées (MH-CR) présentant un terme de contrainte faible et un modèle couplé, et les MH Couplées par Contrainte Faible (MH-CCF) avec deux modèles découplés.

Référence et assimilation découplée

La référence que nous considérons est une MH-C avec un modèle fortement couplé. Nous considérerons donc notre modèle de diffusion fortement couplé (à convergence de Schwarz) au sein d'un 4D-Var incrémental, en contrôlant l'état initial du modèle. Nous nommerons cette méthode MH-C-F (F pour fortement couplée). L'utilisation d'un modèle fortement couplé dans la boucle externe et interne de l'algorithme devrait nous fournir la meilleure analyse en terme de qualité de couplage et de restitution de l'état de référence. A l'opposé, nous pouvons considérer la méthode découplée (nommée UNC pour *uncoupled*) qui assimile les données sur l'océan et l'atmosphère de manière indépendante. Cette méthode UNC devrait nous fournir une référence en terme de coût numérique. La tableau 5.1 résume les caractéristiques principales de ces deux méthodes.

Expérience	Algorithme	Modèle BE	Modèle BI	# BE
MH-C-F	MH-C	M^{cvg}	\mathbf{M}^{cvg}	1
UNC	4D-Var	M^{unc}	\mathbf{M}^{unc}	1

Tableau 5.1 – Récapitulatif des méthodes d'assimilation couplée de référence en terme de qualité de l'analyse et de coût de calcul

Les résultats de MH-C-F sont proposés dans le tableau 5.2. Comme attendu, nous pouvons apprécier la qualité de l'analyse fournie par la méthode. En effet, elle présente une erreur de couplage de l'ordre de 10^{-12} et sa RMSE a quasiment été divisée par 15 par rapport à l'ébauche. Cependant cette méthode présente un coût de calcul élevé, 29.9 fois plus élevé que la méthode découplée, dû à la convergence du modèle. (environ 50 itérations de Schwarz pour chaque intégration du modèle). À l'inverse, la méthode découplée présente un coût numérique très faible, mais une qualité d'analyse très faible également, qui est même dégradée par rapport à l'ébauche. Cette dégradation est dû au fait que le forçage appliqué à l'interface des modèles découplés est celui de l'ébauche. La méthode d'assimilation doit donc compenser cette erreur modèle importante afin d'arriver à coller aux observations en fin de fenêtre temporelle. L'objectif utopique des méthodes d'assimilation de données couplée serait d'arriver à produire une analyse de qualité équivalent au MH-C-F, avec un coût de calcul équivalent à UNC. Étant conscient qu'un tel objectif est irréalisable, nous essaierons simplement de nous en rapprocher.

Expérience	# itérations de minimisation	Coût / UNC	ECI	RMSE (°C)
Ébauche	-	-	$7.74 \cdot 10^{-11}$	3.231
MH-C-F	14	29.9	$5.72 \cdot 10^{-12}$	0.216
UNC	22	1	$1.37 \cdot 10^4$	8.338

Tableau 5.2 – Résultats de l'expérience UNC.

Expériences envisagées

Plusieurs angles d'attaque sont possible pour essayer d'atteindre l'objectif cité précédemment. Une première solution envisageable est d'utiliser un modèle faiblement couplé au sein d'un MH-C. Pour cela, nous utilisons notre modèle couplé stoppé après deux itérations de Schwarz, et nous noterons cette expérience MH-C-2. Afin de renforcer le couplage, nous pouvons également contrôler l'interface en plus de la condition initiale, ce qui nous amène à l'expérience MH-C- Γ -2.

Toujours dans l'objectif de créer une analyse plus fortement couplée à l'interface, nous envisagerons le rajout d'une contrainte faible de couplage avec l'expérience MH-CR- Γ -2. Rajouter une contrainte dans la fonction coût augmente cependant la complexité du problème, et donc le coût numérique de l'algorithme. Nous envisagerons donc de le réduire en stoppant les itérations de Schwarz après une seule itération dans l'expérience MH-CR- Γ -1.

Pour réduire encore plus le coût numérique de nos méthodes, nous supprimerons ensuite le couplage au sein du modèle direct. L'analyse sera alors uniquement couplée via la contrainte faible imposée par J^s . Nous pourrions alors envisager différents poids de J^s avec trois expériences : MH-CCF- Γ -1, MH-CCF- Γ -0.1 et MH-CCF- Γ -0.01.

Enfin, nous analyserons les performances des méthodes inhomogènes couplées avec trois expériences : MI-C-F-2BE, MI-C-2-2BE et MI-C-2-S-2BE ayant chacune deux itérations de boucles externes. La première considère le modèle fortement couplé en boucle externe et la seconde le modèle faiblement couplé avec deux itérations de Schwarz. L'expérience MI-C-2-S-2BE diffère de MI-C-2-2BE dans le sens où le first-guess du modèle x_{alr}^0 est donné par la solution calculée dans la boucle externe précédente. Nous réalisons alors un couplage des modèles par itérations de Schwarz avec les itérations de boucles externes (d'où le "S" du nom de la mé-

thode, pour Schwarz). Les caractéristiques de chacune de ces expériences sont récapitulées dans le tableau 5.3. Lorsque le terme J^s est présent dans la fonction coût de l'expérience, nous donnons la valeur de son poids γ considéré.

Remarque 5.2.3

L'expérience MI-C-2-2BE se rapproche du système CERA Laloyaux *et al.* (2015). Nous ne pourrions pas réaliser ici exactement l'algorithme CERA qui nécessite un couplage asynchrone par fenêtres temporelles. Nous proposons cependant un algorithme équivalent avec un modèle faiblement couplé : $M^{k_{\max}}$ avec $k_{\max} = 2$. Les deux modèles d'océan et d'atmosphère sont ensuite découplés au sein de la boucle interne comme dans le système CERA. Nous espérons ainsi pouvoir comparer notre expérience, dans une certaine mesure, à un système d'assimilation couplé existant.

Expérience	Algorithme	# BE	Modèle BE	Modèle BI	γ
MH-C-F	MH-C	1	M^{cvg}	\mathbf{M}^{cvg}	-
MH-C-2	MH-C	1	$M^{k_{\max}}, k_{\max} = 2$	$\mathbf{M}^{k_{\max}}$	-
MH-C- Γ -2	MH-C- Γ	1	$M^{k_{\max}}, k_{\max} = 2$	$\mathbf{M}^{k_{\max}}$	-
MH-CR- Γ -2	MH-CR- Γ	1	$M^{k_{\max}}, k_{\max} = 2$	$\mathbf{M}^{k_{\max}}$	0.1
MH-CR- Γ -1	MH-CR- Γ	1	$M^{k_{\max}}, k_{\max} = 1$	$\mathbf{M}^{k_{\max}}$	0.1
MH-CCF- Γ -1	MH-CCF- Γ	1	M^{unc}	\mathbf{M}^{unc}	1
MH-CCF- Γ -0.1	MH-CCF- Γ	1	M^{unc}	\mathbf{M}^{unc}	0.1
MH-CCF- Γ -0.01	MH-CCF- Γ	1	M^{unc}	\mathbf{M}^{unc}	0.01
MI-C-F-2BE	MI-C	2	M^{cvg}	\mathbf{M}^{unc}	-
MI-C-2-2BE	MI-C	2	$M^{k_{\max}}, k_{\max} = 2$	\mathbf{M}^{unc}	-
MI-C-2-S-2BE	MI-C	2	$M^{k_{\max}}, k_{\max} = 2$	\mathbf{M}^{unc}	-
UNC	4D-Var	1	M^{unc}	\mathbf{M}^{unc}	-

Tableau 5.3 – Récapitulatif des expériences réalisées avec le modèle de diffusion 1D. Pour chaque expérience nous décrivons l'algorithme utilisé, le modèle de la boucle externe (BE), celui de la boucle interne (BI), et la valeur du coefficient de la contrainte faible. Nous mentionnons également le nombre de boucles externes qui sera réalisé.

5.2.3 Résultats

Dans un premier temps, nous irons à convergence de minimisation dans les expériences afin d'évaluer les performances de chaque algorithme à l'optimum. L'ensemble des résultats à convergence est regroupé dans le tableau 5.4. Dans un second temps, nous limiterons le nombre d'itérations de boucles internes pour comparer certaines méthodes à coût de calcul équivalent. Ces second résultats sont résumés dans le tableau 5.5.

Remarque 5.2.4 - À propos des résultats des méthodes inhomogènes

Il nous faut remarquer que cette configuration ne permet pas aux méthodes inhomogènes d'exploiter leur plein potentiel. En effet, les interactions entre les deux modèles ne sont possibles qu'au sein de la boucle externe. Ainsi, en ne considérant qu'une seule fenêtre temporelle, nous limitons fortement les échanges d'informations entre les deux milieux. En effet, une intégration du modèle couplé sur plusieurs sous-fenêtres impliquerait des échanges à chaque changement de sous-fenêtre temporelle.

Résultats à convergence de minimisation

Expérience	# itérations de minimisation	Coût / UNC	ECI	RMSE (°C)
Ébauche	-	-	$7.74 \cdot 10^{-11}$	3.231
MH-C-F	14	29.8	$5.72 \cdot 10^{-12}$	0.216
MH-C-2	14	1.3	8.11	0.416
MH-C- Γ -2	20	1.8	7.19	0.278
MH-CR- Γ -2	184	15.8	$7.06 \cdot 10^{-9}$	0.217
MH-CR- Γ -1	117	5	$6.97 \cdot 10^{-9}$	0.217
MH-CCF- Γ -1	365	15.6	$1.76 \cdot 10^{-8}$	0.286
MH-CCF- Γ -0.1	396	16.9	$1.28 \cdot 10^{-6}$	0.228
MH-CCF- Γ -0.01	390	16.7	$1.15 \cdot 10^{-4}$	0.226
MI-C-F-2BE	29	3	$9.56 \cdot 10^{-12}$	0.571
MI-C-2-2BE	29	1.4	4.96	0.687
MI-C-2-S-2BE	29	1.4	0.60	0.725
UNC	22	1	$1.37 \cdot 10^4$	8.338

Tableau 5.4 – Récapitulatif des résultats des différents algorithmes d'assimilation de données considérés avec le modèle de diffusion 1D.

Avec une méthode homogène couplée. La MH-C-2 présente une qualité de couplage faible avec un ECI à 8.11. De même, sa RMSE est dégradée par rapport à la MH-C-F, attestant de l'impact du couplage du modèle direct sur la qualité de l'analyse. Nous pouvons cependant apprécier le coût de calcul très faible de cette méthode. Cette expérience nous montre finalement qu'utiliser un modèle faiblement couplé au sein d'un 4D-Var classique ne permet pas de proposer une analyse présentant une bonne consistance à l'interface.

Contrôle de l'interface. Une première solution envisagée pour améliorer les performances du MH-C-2 tout en gardant un coût de calcul raisonnable est de contrôler le first-guess l'interface. Nous pouvons alors espérer que le système modifiera ce first-guess dans la bonne direction, afin d'améliorer le couplage et la RMSE. La MH-C- Γ -2 présente en effet des améliorations par rapport au MH-C-2. L'ECI est très légèrement amélioré (de 8.11 à 7.19) et la RMSE a quasiment été divisée par 2. Le coût de calcul est quant à lui resté raisonnable, même s'il a un peu augmenté dû à la plus grande difficulté pour l'algorithme à converger. Le contrôle de l'interface a donc eu un impact bénéfique sur la qualité de l'analyse.

Pénalisation de l'interface. Le seul contrôle de l'interface ne permet tout de même pas d'obtenir des résultats satisfaisants à moindre coût. Nous pouvons alors, en plus de contrôler l'interface, pénaliser cette dernière avec le terme de contrainte faible J^s . Nous observons l'impact bénéfique de cette contrainte avec l'expérience MH-CR- Γ -2 vis à vis de MH-C- Γ -2. En effet, l'ajout de J^s dans la fonction coût a nettement amélioré l'ECI (de 7.19 à $7.06 \cdot 10^{-9}$), ainsi que la RMSE qui devient équivalente à celle du MH-C, notre référence. Enfin, la MH-CR- Γ -1 procure également de très bons résultats, équivalents à ceux de MH-CR- Γ -2, pour un coût de calcul bien

moindre (plus de trois fois plus élevé pour MH-CR- Γ -2). L'ajout d'un terme de pénalisation de l'interface pour des modèles faiblement couplés a donc permis de créer une analyse de qualité à un coût de calcul raisonnable.

Avec deux modèles découplés. Toujours dans l'optique de réduire le coût numérique des algorithmes, nous avons tenté de supprimer complètement le couplage direct entre les deux modèles, et de l'assurer via la contrainte faible imposée par J^s au cours de la minimisation. Les MH-CCF- Γ proposent cela avec des résultats encourageants. En effet, nous pouvons observer une erreur de couplage de l'ordre de 10^{-6} avec la MH-CCF- Γ -0.01 et une RMSE de 0.228 °C. Bien que nous ne parvenons pas à d'aussi bons résultats que la référence, la méthode est tout de même performante. En effet, l'avantage non négligeable est que le couplage est totalement absent du modèle, réduisant ainsi la complexité d'implémentation et les coûts de calculs dus à celui-ci. Cependant, nous observons ici un problème de convergence de la méthode, qui nécessite beaucoup plus d'itérations de minimisations que les autres, impliquant alors un coût de calcul élevé (15 à 16 fois plus élevé que pour la méthode découplée).

Performance des méthodes inhomogènes. Pour conclure ces premiers tests, nous avons également testé la performance d'algorithmes inhomogènes avec ce modèle. Lorsque le modèle fortement couplé est utilisé en boucle externe (MI-C-F-2BE), nous obtenons un bon ECI, ce qui est logique, mais la RMSE n'est pas très bonne. La qualité du couplage se dégrade si nous considérons un modèle faiblement couplé en boucle externe, ainsi que la RMSE, avec les expériences MI-C-2-2BE et MI-C-2-S-2BE. L'ECI a cependant été divisé d'un facteur 10 entre les deux expériences, grâce aux itérations de Schwarz réalisées avec les itérations de boucles externes. Bien que leurs performances soient en deçà des autres expériences, ces méthodes inhomogènes présentent l'avantage d'être peu gourmandes en coût de calcul.

Comparaison d'algorithmes pour un coût de calcul équivalent

En pratique, avec les modèles de prévision opérationnels, nous n'allons pas à convergence de minimisation lors de l'assimilation de données. Nous proposons donc ici de comparer quelques méthodes à un coût de calcul équivalent, à savoir les algorithmes MH-CR- Γ -1, MH-CCF- Γ -0.1, MI-C-2 et MI-C-2-S. Les résultats de cette comparaison sont donnés dans le tableau 5.5. L'expérience MH-CR- Γ -1 est très robuste à la contrainte de coût de calcul puisqu'elle présente une erreur de couplage de l'ordre de 10^{-4} et une RMSE de 0.353. En revanche, la méthode MH-CCF- Γ -0.1 n'est pas très performante sous cette contrainte de coût. Cette méthode nécessite donc plus d'itérations de minimisation pour produire une analyse de qualité raisonnable dans cette configuration. Enfin, les méthodes inhomogènes ne semblent pas trop impactées par cette contrainte. Ces résultats sont logiques vis à vis du nombre d'itérations de minimisation que nécessitaient ces différentes méthodes à convergence. En effet, la MH-CCF- Γ -0.1 nécessite 18 fois plus d'itérations de minimisation pour atteindre sa convergence, la MH-CR- Γ -1 5 fois plus, et les MI 1.45 fois plus. Il est donc normal que ces dernières soient moins impactées par la contrainte, et que la MH-CCF- Γ -0.1 soit la plus sensible à ce critère.

Expérience	# itérations de minimisation	Coût / UNC	ECI	RMSE (°C)
Ébauche	-	-	$7.74 \cdot 10^{-11}$	3.231
MH-CR- Γ -1	22	1	$5.42 \cdot 10^{-4}$	0.353
MH-CCF- Γ -0.1	22	1	$2.98 \cdot 10^3$	12.509
MI-C-2-2BE	20	1	5.0	0.682
MI-C-2-S-2BE	20	1	0.58	0.720
UNC	22	1	$1.37 \cdot 10^4$	8.338

Tableau 5.5 – Récapitulatif des résultats de différents algorithmes d'assimilation pour un coût de calcul équivalent avec le modèle de diffusion 1D.

5.2.4 Convergence des méthodes inhomogènes

Les expériences utilisant des méthodes inhomogènes présentent de plus faibles résultats à convergence de minimisation avec deux boucles externes. Cependant, nous pouvons nous demander si réitérer d'autres boucles externes n'améliorerait pas leurs résultats. En effet, il est tout à fait probable que deux boucles externes ne suffisent pas à atteindre la convergence de l'algorithme, s'il converge. Nous proposons ici d'étudier numériquement la convergence de ces méthodes puis d'observer l'impact du rajout d'une contrainte faible de couplage sur la convergence à partir des résultats du chapitre précédent (théorèmes 4.2.3 et 4.2.4).

Avec notre modèle simple colonne de diffusion

Nous proposons de réaliser une dizaine de boucles externes afin de vérifier expérimentalement si le MI-C converge avec notre système. Nous nommerons ces expériences supplémentaires MI-C-F-10BE, MI-C-2-10BE et MI-C-2-S-10BE. Un premier diagnostic est donné dans le tableau 5.6, où nous pouvons remarquer la nette amélioration des résultats par rapport à ceux où seules deux boucles externes étaient réalisées (voir tableau 5.4). La figure 5.3 met également en avant la convergence de ces méthodes vers les solutions des méthodes homogènes. En effet, l'expérience MI-C-F-10BE semble converger vers les résultats de l'expérience MH-C-F, c'est à dire la solution de sa boucle externe. De même, la MI-C-2-10BE converge vers la solution de MH-C-2. Cette convergence est soumise à une erreur que nous avons pu quantifier dans le chapitre précédent (théorème 4.2.4), d'où le biais observé entre les valeurs des fonctions coûts. Le cas de l'expérience MI-C-2-S-10BE est intéressant puisqu'en réalisant un couplage par itérations de Schwarz avec les boucles externes, nous observons que la méthode converge vers la solution fortement couplée du MH-C-F, et ce grâce à un coût de calcul diminué par deux par rapport à la MI-C-F-10BE.

Expérience	# itérations de minimisation	Coût / UNC	ECI	RMSE (°C)
Ébauche	-	-	$7.74 \cdot 10^{-11}$	3.231
MH-C-F	14	29.9	$5.72 \cdot 10^{-12}$	0.216
MH-C-2	14	1.3	8.11	0.416
MI-C-F-10BE	145	14.9	$7.57 \cdot 10^{-12}$	0.272
MI-C-2-10BE	145	6.9	10.32	0.500
MI-C-2-S-10BE	149	7	$1.22 \cdot 10^{-3}$	0.277
UNC	22	1	$1.37 \cdot 10^4$	8.338

Tableau 5.6 – Résultats des méthodes inhomogènes avec 10 boucles externes.

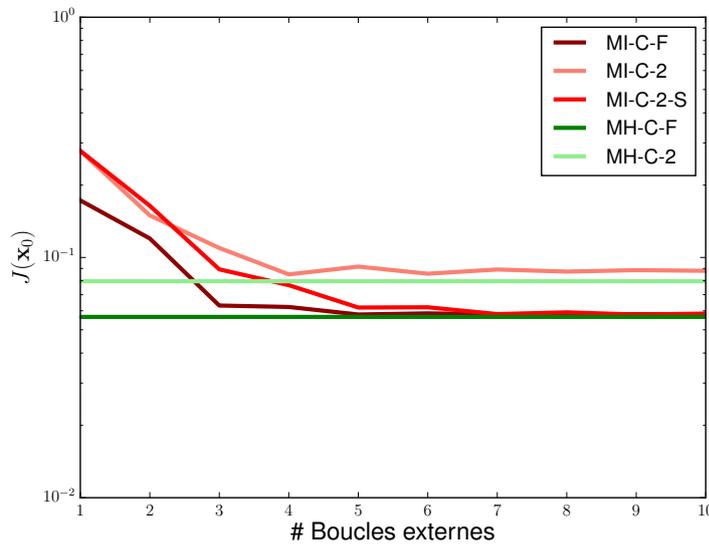


FIGURE 5.3 – Évolution de la fonction coût en fonction du nombre de boucles externes des méthodes inhomogènes couplées par rapport aux méthodes homogènes couplées.

Nous pouvons également illustrer la convergence des MI-C avec la figure 5.4. Nous y représentons l'évolution de l'analyse \mathbf{x}_0 évaluée par les trois algorithmes inhomogènes pour chaque boucle externe. On observe ainsi que la solution évaluée après une boucle externe est identique pour les trois algorithmes, puis la dynamique du modèle couplé entre en jeu dès la seconde itération. En effet, la méthode MI-C-F se démarque par rapport aux autres et semble alors plus performante proche de l'interface côté atmosphère. Puis, nous observons l'impact de la mise à jour de l'interface dans la méthode MI-C-2-S à partir de la troisième itération, où la solution de cet algorithme rejoint la solution de la MI-C-F. Les deux méthodes convergent alors vers la même solution, comme l'atteste les deux profils quasiment confondus à partir de la 7ème itération. En revanche, la MI-C-2 converge vers une solution de moins bonne qualité proche de l'interface que celle des deux autres méthodes. Ce graphe nous permet donc de nous rendre compte de la vitesse de convergence des MI dans un cadre linéaire, et d'apporter une représentation visuelle aux performances de ces méthodes. Nous pouvons alors apprécier la performance de la MI-C-2-S qui propose une solution équivalente à la MI-C-F, pour un coût de calcul bien moindre.

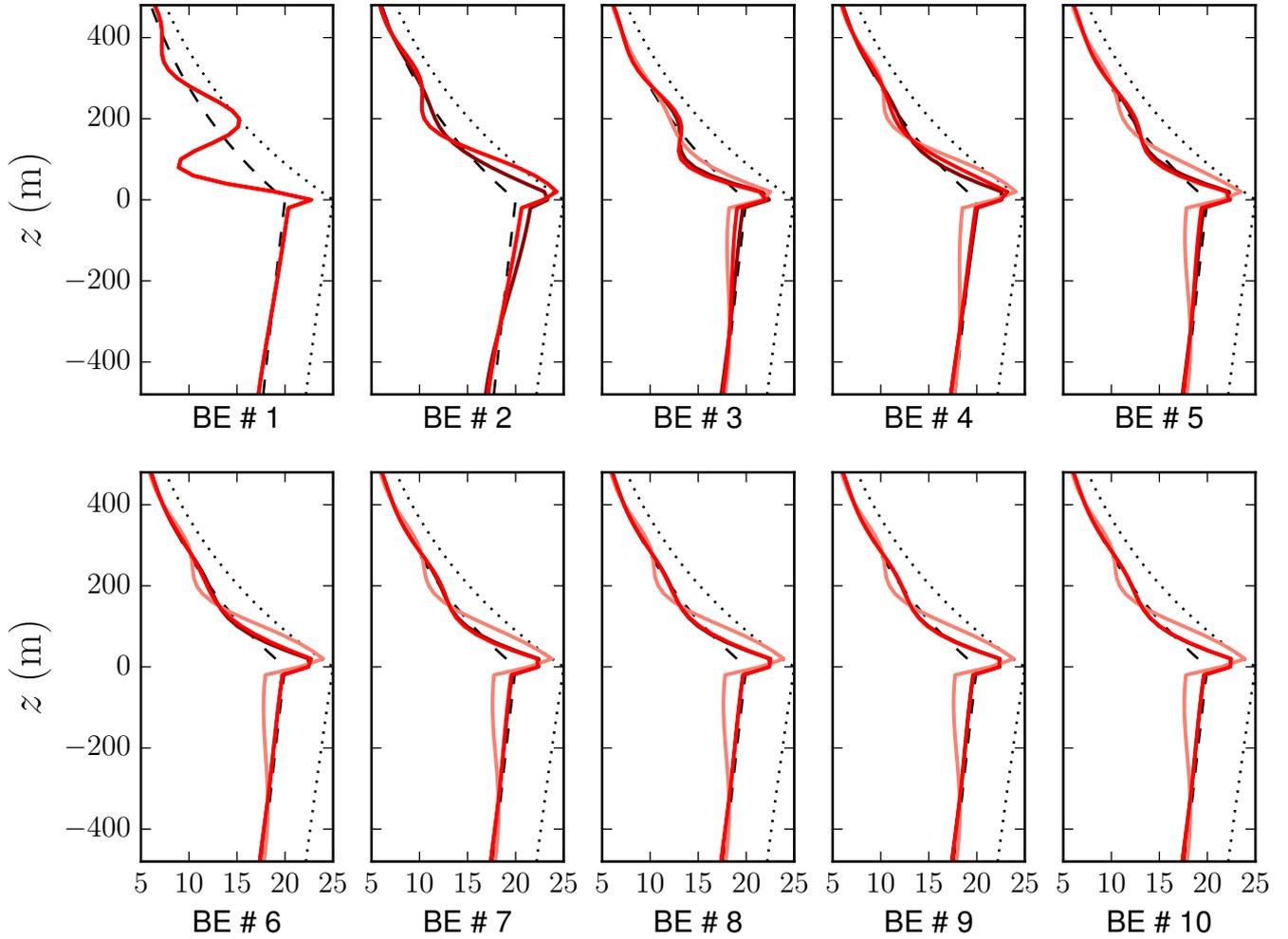


FIGURE 5.4 – Profils du vecteur d'état $\mathbf{x}_0^{(p)}$ évalués par les différentes MI-C en fonction du nombre d'itérations de boucles externes ($p \in \llbracket 1, 10 \rrbracket$). L'état vrai est représenté par le profil en tirets et l'ébauche par celui en pointillée. L'analyse proposée par la MI-C-F est représenté en bordeaux, celle de la MI-C-2 en rose, et celle de la MI-C-2-S en rouge. L'axe vertical est un zoom du domaine Ω , et l'axe des abscisses représente la température en $^{\circ}\text{C}$.

À partir du critère de convergence théorique

Nous venons d'étudier numériquement la convergence des MI. Nous souhaitons également évaluer la convergence de ces algorithmes à partir du critère de convergence obtenu dans le chapitre 4 pour des problèmes linéaires (théorème 4.2.3). Cependant, il est complexe de représenter un pas de temps du modèle couplé $\mathbf{M}_{i,i+1}$ sous forme matricielle, ce dernier étant couplé par un algorithme de Schwarz. Nous considérerons donc une version linéaire simplifiée du modèle couplé afin d'évaluer l'impact de la présence de J^s dans les méthodes inhomogènes à partir de nos théorèmes. Nous définissons $\mathbf{x}_0 = (\mathbf{x}_{a,0}, \mathbf{x}_{o,0}, \mathbf{x}_{a|\Gamma}, \mathbf{x}_{o|\Gamma})^T$ le vecteur de contrôle. Les modèles d'océan \mathbf{M}_o et d'atmosphère \mathbf{M}_a sont définis à partir des équations du modèle précédemment présenté, et les opérateurs d'interface sont, comme précédemment, des opérateurs de Dirichlet et de Neumann. L'intégration sur une fenêtre temporelle est réalisé en 4

étapes :

- couplage de type Dirichlet entre l’atmosphère et l’océan,
- intégration du modèle d’océan et calcul du flux de Neumann à chaque pas de temps ,
- couplage de type Neumann entre l’océan et l’atmosphère,
- intégration du modèle d’atmosphère et calcul du flux de Dirichlet à chaque pas de temps,

puis il est possible de réitérer ce processus pour la fenêtre temporelle suivante. Nous obtenons alors un modèle couplé linéaire \mathbf{M} que nous pouvons écrire sous forme matricielle à partir des différents opérateurs linéaires. De la même manière, nous pouvons exprimer sous forme matricielle le modèle découplé $\tilde{\mathbf{M}}$. A partir de ces deux modèles écrits sous forme matricielle, nous pouvons évaluer si la MI-C converge dans des conditions données (choix du nombre de pas de temps, d’espace, temps d’intégration, etc), et évaluer l’impact de J^s sur la convergence. Considérons dans un premier temps la configuration C1 suivante :

$$\begin{array}{lll} v_a = 1 \text{ m}^2 \cdot \text{s}^{-1} & \Delta t_a = \Delta t_o = 20 \text{ s} & \Delta z_a = 10 \text{ m} \\ v_o = 0.1 \text{ m}^2 \cdot \text{s}^{-1} & N_j = 15 & \Delta z_o = 1 \text{ m} \\ & N_{\text{fen}} = 4 & N_a = N_o = 15 \end{array}$$

où N_j représente le nombre de pas de temps dans chaque fenêtre temporelle $j \in [0, N_{\text{fen}} - 1]$. N_a et N_o représentent le nombres de niveaux verticaux pour chacun des milieux. Nous observons toute la colonne au dernier pas de temps d’intégration, et considérons les matrices $\mathbf{B} = 0.1 \cdot \mathbf{I}$ et $\mathbf{R} = 0.05 \cdot \mathbf{I}$. A partir de cette configuration C1, nous pouvons calculer le critère de convergence de la MI-C, $\|\mathbf{I} - \tilde{\mathbf{K}}\Pi\|_2 = 1.12$. La MI-C ne converge donc pas puisqu’il est nécessaire et suffisant que ce critère soit < 1 pour assurer la convergence, d’après le théorème 4.2.3. Nous évaluons alors l’impact de l’ajout de la contrainte faible de couplage J^s (5.4) dans la fonction coût sur la convergence. Pour cela, nous calculons le critère de convergence d’une MI-CRCF en fonction du coefficient γ . La figure 5.5 présente l’évolution du critère de convergence. Nous observons que l’algorithme converge lorsque $\gamma \in [1, 300]$. L’ajout de la contrainte faible de couplage semble donc permettre la convergence des MI si son poids est judicieusement choisi.

Nous pouvons également nous intéresser à l’évolution de l’erreur d’une MI lorsque J^s est présent ou non. Considérons pour cela la configuration C2 qui diffère de C1 pour le paramètre $\Delta z_a = 15 \text{ m}$. Dans cette configuration, la MI-C converge puisque son critère de convergence est égal à 0.88. Nous pouvons alors évaluer si l’ajout de la contrainte faible de couplage permet d’améliorer l’erreur commise par rapport à la solution optimale. L’erreur est donnée par le théorème 4.2.4 du chapitre précédent : $\tilde{\boldsymbol{\varepsilon}} = \left(\mathbf{K} - (\tilde{\mathbf{K}}\Pi)^{-1} \tilde{\mathbf{K}} \right) \mathbf{d}^{(0)}$. Nous considérons alors des écarts aux observations $\mathbf{d}^{(0)} = \mathbf{1} \text{ }^\circ\text{C}$ ($\mathbf{1}$ représente le vecteur unitaire) sur toute la colonne observée, et nous avons alors $\mathbf{d}^{(0)} = (0 \dots 0, \mathbf{d}^{(0)T})^T$ où les 0 correspondent à la différence à l’ébauche pour la première boucle externe. Nous représentons ensuite l’évolution de $\|\tilde{\boldsymbol{\varepsilon}}\|_2$ en fonction de γ dans la figure 5.6. Nous en déduisons que l’ajout de la contrainte faible, même en cas de convergence du MI-C, a un bénéfice puisque l’on minimise l’erreur commise par rapport à la solution du problème. Plus le poids de J^s est fort, plus l’erreur est faible. Attention cependant à ne pas trop contraindre la fonction coût, puisque cela pourrait à l’inverse dégrader la convergence de l’algorithme.

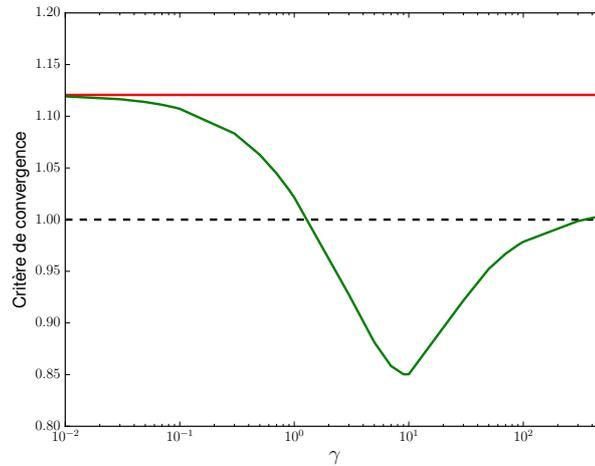


FIGURE 5.5 – Évolution du critère de convergence des MI en fonction du poids de la contrainte faible de couplage J^S . En rouge le critère de convergence d'une MI-C, sans terme J^S , et en vert l'évolution du critère pour une MI-CRCE.

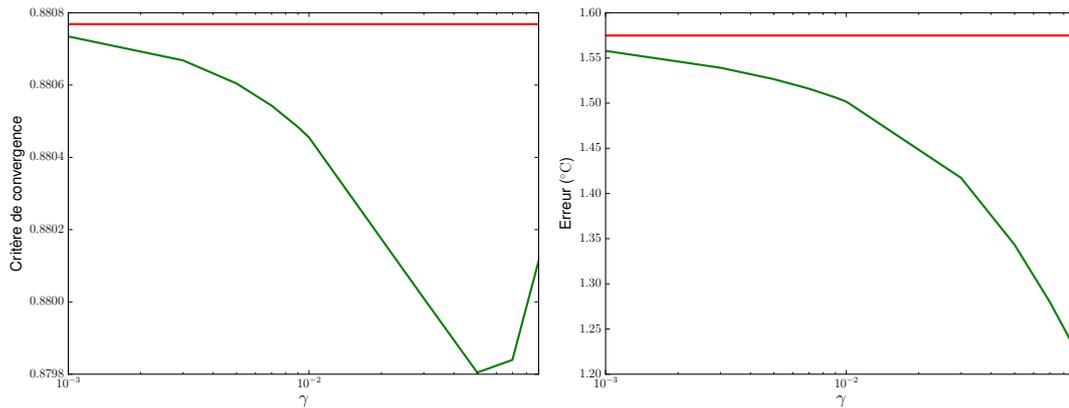


FIGURE 5.6 – À gauche : évolution du critère de convergence des MI en fonction du poids de la contrainte faible de couplage J^S . À droite : évolution de l'erreur par rapport à la solution optimale en fonction du poids de la contrainte faible de couplage J^S . La courbe rouge désigne une MI-C sans terme J^S , et la courbe verte une MI-CRCE.

5.3 Discussions et perspectives

Nous avons pu dans ce chapitre tester plusieurs algorithmes d'assimilation de données que nous avons introduits, dans le cadre d'un système simple composé d'un modèle linéaire de diffusion 1D vertical. Ce cas d'application, bien que très simple, nous a cependant permis de comparer différents algorithmes et d'observer quelques limites à chacun d'entre eux.

- Les méthodes homogènes couplées sont limitées pour produire une analyse de qualité avec un coût de calcul raisonnable. En effet, elles ont le potentiel de produire une analyse de très bonne qualité, avec l'utilisation d'un modèle fortement couplé, mais cela engendre alors un coût de calcul beaucoup trop élevé. Il a alors pu être montré que l'utilisation d'un modèle faiblement couplé au sein de ces méthodes ne permettait pas de

produire une analyse de bonne qualité, malgré un coût de calcul raisonnable.

- Les méthodes homogènes couplées avec un contrôle de l'interface améliorent la qualité de l'analyse par rapport à une MH-C avec un modèle faiblement couplé, mais ne produisent toujours pas de résultats satisfaisants.
- Le rajout d'une contrainte faible de couplage aux méthodes homogènes permet alors de produire une analyse de meilleure qualité mais le coût de calcul en est impacté. Il reste cependant raisonnable dans certaines configurations, et nous parvenons également à produire une analyse de qualité en supprimant le couplage direct entre les modèles. Le coût de calcul est cependant élevé dans ce dernier cas.
- Enfin, les méthodes inhomogènes sont également performantes. Elles permettent à convergence de se rapprocher de la solution d'une méthode homogène pour un faible coût de calcul. De plus, nous avons pu montrer qu'utiliser les itérations de boucles externes comme des itérations de Schwarz pour le modèle couplé permettait même d'aller au delà de la solution de la boucle externe initiale. Cette technique permet de produire une analyse présentant un bon équilibre à l'interface avec un coût de calcul très faible.

Ces premiers tests nous ont permis de commencer à comparer les différents types d'algorithmes d'assimilation de données. Nous observons qu'il est nécessaire de faire un choix entre la qualité d'analyse et le coût de calcul. Nous pouvons tout de même apprécier l'impact positif que peut avoir l'ajout d'une contrainte faible de couplage au cours de la minimisation. En effet, cette contrainte permet alors d'améliorer la consistance de l'analyse à l'interface sans avoir supporter le coût d'un modèle couplé. De même, cette contrainte peut permettre de converger dans le cas des méthodes inhomogènes. Ces méthodes inhomogènes sont également prometteuses, notamment lorsque nous mettons à jour l'interface à chaque itération de boucle externe. Il nous faut cependant garder à l'esprit que nous avons réalisé ces expériences avec un modèle de diffusion 1D très simple. Un modèle opérationnel couplé océan-atmosphère présente d'autres difficultés qui pourraient alors limiter les performances des algorithmes. Une méthode inhomogène ne convergera alors peut-être pas, dû à la trop grande différence entre le modèle couplé et ses composantes découplés. A l'inverse, les forçages et paramétrisations des modèles opérationnels pourraient également masquer des difficultés, et tous les algorithmes pourraient bien se comporter. Nous allons donc, dans la troisième et dernière partie de ce manuscrit, nous intéresser à un cadre d'étude plus complexe, en utilisant un modèle couplé colonne non-linéaire avec des paramétrisations sous-maille proche de l'interface air-mer.

Troisième partie

ÉVALUATION D'ALGORITHMES D'ASSIMILATION AVEC UN MODÈLE COUPLÉ COLONNE AU SEIN D'UN SYSTÈME DE PRÉVISION COMPLEXE

Nous venons dans la partie 2 de ce manuscrit de présenter un certain nombre d'algorithmes répondant au problème de l'assimilation de données couplée. Ces algorithmes ont été testés et évalués avec un modèle de diffusion linéaire à une dimension. Conscient qu'un tel modèle ne peut pas attester sérieusement de la robustesse des différents algorithmes, nous allons considérer un modèle plus complexe et plus représentatif des modèles de prévision opérationnels dans cette troisième et dernière partie.

- Nous commencerons dans le chapitre 6 par développer un modèle couplé non-linéaire 1D. Ce modèle présente des paramétrisations sous-maille proches de l'interface permettant de modéliser les phénomènes turbulents qui y ont lieu. Ses équations découlent des équations primitives présentées dans le chapitre 2. Le couplage à l'interface air-mer est réalisé à partir des formulations bulk et les coefficients de diffusions sont évalués à partir de paramétrisations complexes. Il s'agit donc d'un modèle plus simple qu'un modèle opérationnel, mais représentatif du comportement de ces derniers.
 - Nous implémenterons ensuite ce modèle et les algorithmes d'assimilation couplée au sein d'une surcouche logicielle qui nous a été fourni par le CEPMMT : OOPS. Nous décrirons rapidement ce logiciel dans le chapitre 7, ainsi que les développements que nous lui avons apporté.
 - Enfin, le dernier chapitre de ce manuscrit est consacré aux résultats obtenus à partir des différents algorithmes d'assimilation couplée considérés avec le modèle non-linéaire. Nous aurons alors un meilleur aperçu de la robustesse de chacun des algorithmes et pourrons évaluer les alternatives que nous avons proposées par rapport à des systèmes existants tel que le CERA.
-

Chapitre 6

Un modèle couplé simple colonne avec paramétrisations sous-maille

Sommaire

6.1 Description du modèle couplé colonne	106
6.1.1 Des équations primitives 3D au modèle simple colonne	106
6.1.2 Conditions aux limites et couplage océan-atmosphère	108
6.1.3 Paramétrisation de la couche limite	110
6.1.4 Récapitulatif	111
6.2 Choix d'implémentation du modèle	112
6.2.1 Discrétisation temporelle	112
6.2.2 Définition de la grille de calcul	112
6.2.3 Flux atmosphériques imposés	113
6.2.4 Rappels grande échelle	116
6.3 Stratégies de couplage	116

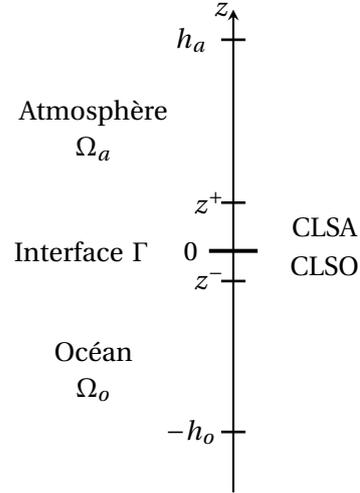
Pour étudier le comportement des modèles à l'interface air-mer en présence des paramétrisations sous-maille, il est possible de se ramener à un cas d'étude simplifié en 1D, tout en gardant la complexité de modélisation des paramétrisations à l'interface. Cela est réalisé en effectuant quelques hypothèses simplificatrices supplémentaires à celles des équations primitives 3D présentées dans le chapitre 2. Le modèle 1D, dit *simple colonne*, sera donc représentatif d'un modèle *réaliste* basé sur ces équations primitives, avec la simplicité et le faible coût numérique d'un modèle 1D. De tels modèles sont souvent appelés modèles *jouets* dans la littérature.

⚠ Un modèle simple colonne n'a pas pour but de faire de la prévision météorologique mais se comporte de manière semblable à un modèle réaliste 3D : nous y retrouverons les principales difficultés de modélisation à l'interface air-mer. C'est pourquoi les modèles simple colonne sont très utilisés pour étudier les paramétrisations sous-maille dans la couche limite de surface.

Nous allons dans ce chapitre décrire un modèle simple colonne couplé océan-atmosphère que nous utiliserons dans la suite pour tester la robustesse de nos algorithmes d'assimilation couplée. Nous commencerons par présenter les équations du modèle avant de définir plusieurs choix d'implémentation numérique.

6.1 Description du modèle couplé colonne

Tout d'abord, définissons le domaine spatio-temporel du modèle. Le domaine atmosphérique sera désigné par $\Omega_a = [0, h_a]$ et le domaine océanique par $\Omega_o = [-h_o, 0]$. La couche limite de surface est comprise entre les coordonnées z^+ dans l'atmosphère et z^- dans l'océan. L'interface océan atmosphère est représentée par Γ . La figure ci-contre récapitule cela. Enfin, le modèle évoluera sur la fenêtre temporelle $[0, T]$ avec $T > 0$ s.



6.1.1 Des équations primitives 3D au modèle simple colonne

Les équations de notre modèle simple colonne dérivent des équations primitives 3D (2.1), (2.2), (2.3), (2.4) et (2.5) présentées dans le chapitre 2, auxquelles nous ajoutons plusieurs hypothèses simplificatrices pour se ramener à un modèle 1D. Nous allons commencer par décrire ces hypothèses afin de présenter les équations du modèle.

Hypothèses simple colonne et décomposition de Reynolds

La première hypothèse que nous ferons est de considérer l'homogénéité horizontale :

$$\frac{\partial \cdot}{\partial x} = \frac{\partial \cdot}{\partial y} = 0$$

Cette hypothèse est raisonnable dans le sens où nous nous intéressons à la dynamique du couplage océan-atmosphère. Or cette dynamique est dominée par les turbulences proches de l'interface qui fluctuent principalement selon la verticale. Nous simplifions les équations primitives 3D à partir de cette hypothèse pour obtenir les équations suivantes :

$$\begin{aligned} \text{(Équation de quantité de mouvement)} \quad & \frac{\partial \mathbf{u}}{\partial t} = -f_C \mathbf{k} \times \mathbf{u} + \frac{\partial w \mathbf{u}}{\partial z} + \nu_m \frac{\partial^2 \mathbf{u}}{\partial z^2} + f_{\mathbf{u}} \\ \text{(Équation de continuité)} \quad & \frac{\partial w}{\partial z} = 0 \\ \text{(Équation de conservation des traceurs)} \quad & \frac{\partial \mathcal{T}}{\partial t} = \frac{\partial w \mathcal{T}}{\partial z} + \nu_s \frac{\partial^2 \mathcal{T}}{\partial z^2} + f_{\mathcal{T}} \end{aligned}$$

En appliquant ensuite la décomposition de Reynolds à ce système d'équations, comme cela a été présenté dans le chapitre 2, nous obtenons :

$$\begin{aligned}\frac{\partial \langle \mathbf{u} \rangle}{\partial t} &= -f_C \mathbf{k} \times \langle \mathbf{u} \rangle + \frac{\partial \langle w \rangle \langle \mathbf{u} \rangle}{\partial z} + \frac{\partial \langle w' \mathbf{u}' \rangle}{\partial z} + \nu_m \frac{\partial^2 \langle \mathbf{u} \rangle}{\partial z^2} + f_{\mathbf{u}} \\ \frac{\partial \langle \mathcal{T} \rangle}{\partial t} &= \frac{\partial \langle w \rangle \langle \mathcal{T} \rangle}{\partial z} + \frac{\partial \langle w' \mathcal{T}' \rangle}{\partial z} + \nu_s \frac{\partial^2 \langle \mathcal{T} \rangle}{\partial z^2} + f_{\mathcal{T}} \\ \frac{\partial \langle w \rangle}{\partial z} &= 0\end{aligned}$$

La dynamique proche de l'interface est principalement dominée par les échelles sous-maille. Notre seconde hypothèse sera alors de négliger les termes liés à l'advection verticale résolue, permettant de simplifier un peu plus nos équations. Les fluctuations sous-maille non résolues $\langle w' \mathbf{u}' \rangle$ et $\langle w' \mathcal{T}' \rangle$ sont quant à elles exprimées grâce à l'hypothèse de fermeture turbulente introduite dans le chapitre 2 :

$$\begin{aligned}\langle w' \mathbf{u}' \rangle &= K_m \frac{\partial \langle \mathbf{u} \rangle}{\partial z} \\ \langle w' \mathcal{T}' \rangle &= K_s \frac{\partial \langle \mathcal{T} \rangle}{\partial z}\end{aligned}$$

où K_m est le coefficient de viscosité turbulente et K_s est le coefficient de diffusivité turbulente. Par soucis de concision, comme dans le chapitre 2, nous nous affranchirons dans la suite de la notation $\langle \cdot \rangle$.

Équations du modèle 1D simple colonne

A partir des hypothèses citées précédemment, nous obtenons les équations de notre modèle 1D simple colonne :

$$\begin{aligned}\frac{\partial \mathbf{u}_\alpha(z, t)}{\partial t} &= -f_C \mathbf{k} \times \mathbf{u}_\alpha(z, t) + \frac{\partial}{\partial z} \left(K_m^\alpha(z) \frac{\partial \mathbf{u}_\alpha(z, t)}{\partial z} \right) + f_{\mathbf{u}_\alpha}(z, t) \quad \text{sur } \Omega_\alpha \times [0, T] \\ \frac{\partial \mathcal{T}_\alpha(z, t)}{\partial t} &= \frac{\partial}{\partial z} \left(K_s^\alpha(z) \frac{\partial \mathcal{T}_\alpha(z, t)}{\partial z} \right) + f_{\mathcal{T}_\alpha}(z, t) \quad \text{sur } \Omega_\alpha \times [0, T]\end{aligned} \tag{6.1}$$

avec $\alpha = a, o$ désignant les variables atmosphériques ou océaniques. Les deux modèles d'atmosphère et d'océan ont ainsi la même structure et se différencieront par le choix des seconds membres, des conditions d'interface et le calcul des coefficients de viscosité et de diffusivité turbulente.

Remarque 6.1.1

On remarquera que les termes de viscosité et de diffusion moléculaires (ν_m et ν_s) ont disparu. Cela vient du fait qu'ils sont négligeables par rapport aux termes turbulents (K_m et K_s) dans la couche limite de surface. En dehors des couches limites, nous considérerons les coefficients de turbulence comme étant constants et égaux aux coefficients moléculaires, représentant alors l'atmosphère libre et l'océan intérieur correctement.

Les équations (6.1) sont complétées par les second membres $f_{\mathbf{u}_{a,o}}(z, t)$ et $f_{\mathcal{T}_{a,o}}(z, t)$ suivants :

$$\begin{aligned} \forall (z, t) \in \Omega_{a,o} \times [0, T] : \\ f_{\mathbf{u}_a}(z, t) &= -f_C \mathbf{k} \times \mathbf{u}_{ag}(z, t) && \text{rappel vers les vents géostrophiques } \mathbf{u}_{ag}(z, t) \\ f_{\theta_a}(z, t) &= \lambda_S(z) (\theta_a(z, t) - \theta_{aLS}(z, t)) && \text{rappel grande échelle vers } \theta_{aLS}(z, t) \\ f_{q_a}(z, t) &= \lambda_S(z) (q_a(z, t) - q_{aLS}(z, t)) && \text{rappel grande échelle vers } q_{aLS}(z, t) \\ f_{\mathbf{u}_o}(z, t) &= 0 \\ f_{\theta_o}(z, t) &= \partial_z \left(\frac{Q_S(z, t)}{\rho_o C_p^o} \right) && \text{forçage par le flux solaire pénétrant } Q_S(z, t) \\ f_{S_o}(z, t) &= 0 \end{aligned}$$

avec $1/\lambda_S(z)$ un temps de relaxation variable avec l'altitude permettant de définir le coefficient de nudging pour les termes de rappel grande échelle atmosphérique et C_p^o la capacité thermique de l'eau. $Q_S(z, t)$ est le flux solaire pénétrant dans l'océan, calculé à partir du flux solaire $Q_{S_0} = Q_S(0, t)$ arrivant à la surface de l'océan.

Remarque 6.1.2

Les rappels grande échelle (\mathbf{u}_{ag} , θ_{aLS} et q_{aLS}), λ_S , ainsi que le flux solaire arrivant en surface Q_{S_0} seront à définir lorsque nous implémenterons numériquement le modèle.

6.1.2 Conditions aux limites et couplage océan-atmosphère

Pour compléter la description du modèle, il nous faut définir les conditions aux limites extérieures et à l'interface océan-atmosphère du modèle. Ces dernières nous permettront d'imposer la consistance des flux turbulents à l'interface et introduiront alors le problème du couplage dans le modèle, comme cela a été présenté par les équations (2.8) et (2.9) dans le chapitre 2.

Conditions aux limites extérieures

Les équations du modèle (6.1) sont complétées par des conditions aux limites en $z = h_a$ et $z = -h_o$. Nous noterons dans la suite $\partial\Omega_\alpha^{\text{ext}}$ ($\alpha = a, o$) les bords extérieurs des domaines atmosphériques et océaniques. Nous considérons alors les conditions sur $\partial\Omega_\alpha^{\text{ext}}$ suivantes :

$$\begin{aligned} \frac{\partial \mathbf{u}_\alpha}{\partial z}(t) &= 0 && \text{sur } \partial\Omega_\alpha^{\text{ext}} \\ \frac{\partial \mathcal{T}_\alpha}{\partial z}(t) &= \mathcal{F}_{\mathcal{T}_\alpha}^{\text{ext}} && \text{sur } \partial\Omega_\alpha^{\text{ext}} \end{aligned}$$

où $\mathcal{F}_{\mathcal{T}_\alpha}^{\text{ext}}$ est le flux imposé au traceur \mathcal{T}_α à la frontière extérieure du domaine tel que :

$$\begin{aligned} \mathcal{F}_{\theta_a}^{\text{ext}} &= 0 && \mathcal{F}_{\theta_o}^{\text{ext}} = \nu_s \cdot \left. \frac{\partial \theta_o(z, 0)}{\partial z} \right|_{z=-h_o} \\ \mathcal{F}_{q_a}^{\text{ext}} &= 0 && \mathcal{F}_{S_o}^{\text{ext}} = \nu_s \cdot \left. \frac{\partial S_o(z, 0)}{\partial z} \right|_{z=-h_o} \end{aligned}$$

Continuité du flux de quantité de mouvement

La continuité du flux de quantité de mouvement à l'interface Γ impose que :

$$\rho_a K_m^a \frac{\partial \mathbf{u}_a}{\partial z} = \rho_o K_m^o \frac{\partial \mathbf{u}_o}{\partial z} = \boldsymbol{\tau} \quad \text{sur } \Gamma \times [0, T]$$

où on note $\boldsymbol{\tau}$ le flux de quantité de mouvement qui peut être exprimé par la formulation bulk :

$$\boldsymbol{\tau} = \rho_a C_d \|\mathbf{u}_{a|_r} - \mathbf{u}_{o|_r}\| (\mathbf{u}_{a|_r} - \mathbf{u}_{o|_r})$$

avec C_d le coefficient de transfert bulk de traînée (\cdot_d pour *drag coefficient*). C_d est calculé via une formulation bulk comme cela a été mentionné dans le chapitre 2.

Continuité du flux de chaleur

Nous utiliserons une formulation classique de la continuité du flux de chaleur à l'interface air-mer des modèles couplés. On définit Q_{net} le flux de chaleur net à l'interface océan-atmosphère tel que :

$$Q_{\text{net}} = Q_{S_0} + Q_{L\downarrow} + Q_{L\uparrow} + Q_E + Q_H$$

avec :

- Q_{S_0} le flux radiatif solaire arrivant en surface de l'océan
- $Q_{L\downarrow}(t)$ les flux radiatifs de grandes longueurs d'ondes autres que le flux solaire Q_{S_0}
- $Q_{L\uparrow}$ le rayonnement émis par l'océan de type corps noir
- Q_H le flux de chaleur sensible
- $Q_E = \Lambda E$ le flux de chaleur latente défini à partir du flux d'évaporation E et de la chaleur latente de vaporisation $\Lambda = 2.5 \cdot 10^6 \text{ J.kg}^{-1}$

Les flux de chaleur sensible Q_H et d'évaporation E sont exprimés par les formulations bulk suivantes :

$$Q_H = \rho_a C_p^a C_H \|\mathbf{u}_{a|_r} - \mathbf{u}_{o|_r}\| (\theta_{a|_r} - \theta_{o|_r})$$

$$E = \rho_a C_E \|\mathbf{u}_{a|_r} - \mathbf{u}_{o|_r}\| (q_{a|_r} - q_{o|_r})$$

avec C_H le coefficient de transfert bulk de chaleur sensible, C_p^a la capacité thermique de l'air et C_E un coefficient d'échange. $q_{o|_r}$ est l'humidité à la surface de l'océan qui peut être évaluée à partir de la température de surface de l'océan en supposant que l'air est saturé à l'interface air-mer :

$$q_{o|_r} = q_{\text{sat}} = \frac{0.98}{\rho_o} 640380 \exp\left(-\frac{5107.4}{\theta_{o|_r}}\right) \quad (\text{Large, 2006b})$$

Le flux de radiation $Q_{L\uparrow}$ est directement donné par le modèle océanique (l'océan étant considéré comme un corps noir) et suit la loi de Stefan-Boltzmann : $Q_{L\uparrow} = -\varepsilon \sigma_S (\theta_{o|_r})^4$ avec $\varepsilon = 1$ l'émissivité de l'océan considérée et $\sigma_S = 5.67 \cdot 10^{-8} \text{ W.m}^{-2}.\text{K}^{-4}$ la constante de Stefan. La continuité du flux de chaleur Q_{net} à l'interface air-mer nous permet tout d'abord d'écrire la condition d'interface à l'océan suivante :

$$K_s^o \frac{\partial \theta_o}{\partial z} + \frac{Q_{S_0}}{\rho_o C_p^o} = \frac{Q_{\text{net}}}{\rho_o C_p^o} \quad \text{sur } \Gamma \times [0, T]$$

avec C_p^o la capacité thermique de l'eau. Nous ferons ensuite l'hypothèse d'une atmosphère transparente pour définir ses conditions d'interface. Cette hypothèse consiste à considérer que les flux radiatifs ne sont pas absorbés par l'atmosphère. La continuité du flux de chaleur net est alors complétée par les deux conditions d'interface de l'atmosphère suivantes :

$$K_s^a \frac{\partial \theta_a}{\partial z} = \frac{Q_H}{\rho_a C_p^a} \quad \text{sur } \Gamma \times [0, T]$$

$$K_s^a \frac{\partial q_a}{\partial z} = \frac{-E}{\rho_a} \quad \text{sur } \Gamma \times [0, T]$$

Flux d'eau douce

Enfin, il nous reste à définir la condition à l'interface Γ du traceur de salinité S_o . On supposera que nous avons la condition d'interface suivante :

$$K_s^o \frac{\partial S_o}{\partial z} = F$$

où F est le flux d'eau douce pénétrant dans l'océan, défini tel que :

$$F = 0.001(E - P)S_{o|\Gamma}$$

avec P la quantité de précipitations provenant de l'atmosphère.

Remarque 6.1.3

E et P sont usuellement exprimés en $\text{kg.m}^{-2}.\text{s}^{-1}$, or le flux d'eau douce F pénétrant dans l'océan doit être exprimé en m.s^{-1} . C'est pourquoi nous multiplions $(E - P)$ par la constante 0.001, du fait qu'un flux de $1 \text{ kg.m}^{-2}.\text{s}^{-1}$ d'eau équivaut à 0.001 m.s^{-1} .

Récapitulatif des différents flux

Pour résumer, nous avons exprimé les conditions aux limites d'interface de notre modèle grâce à l'expression de différents flux :

Flux de quantité de mouvement	$\boldsymbol{\tau} = \rho_a C_d \ \mathbf{u}_{a \Gamma} - \mathbf{u}_{o \Gamma}\ (\mathbf{u}_{a \Gamma} - \mathbf{u}_{o \Gamma})$
Flux de chaleur sensible	$Q_H = \rho_a C_p^a \ \mathbf{u}_{a \Gamma} - \mathbf{u}_{o \Gamma}\ (\theta_{a \Gamma} - \theta_{o \Gamma})$
Flux d'évaporation	$E = \rho_a C_E \ \mathbf{u}_{a \Gamma} - \mathbf{u}_{o \Gamma}\ (q_{a \Gamma} - q_{\text{sat}})$
Flux de chaleur latente	$Q_E = \Lambda E$
Flux de précipitations	P
Flux d'eau douce	F
Flux radiatif solaire	Q_{S_0}
Flux radiatifs	$Q_{L\uparrow}, Q_{L\downarrow}$
Flux de chaleur net	$Q_{\text{net}} = (1 - \alpha)Q_{S_0} + Q_{L\downarrow} + Q_{L\uparrow} + Q_E + Q_H$

Remarque 6.1.4

Les flux de précipitations, solaire et de radiation $Q_{L\downarrow}$ sont normalement calculés et fournis par le modèle atmosphérique. Notre modèle étant un modèle colonne simplifié, ils ne seront pas calculés par ce dernier mais directement imposés.

6.1.3 Paramétrisation de la couche limite

Pour terminer la description du modèle, il nous faut expliciter le calcul des coefficients de viscosité K_s^a et de diffusion turbulente K_m^a ($a = a, o$) au sein de la couche limite. Ce calcul n'est pas trivial et nous ne le présenterons pas dans les détails dans ce manuscrit. En ce qui concerne les coefficients de viscosité et de diffusion turbulente dans l'océan (K_s^o et K_m^o), ils seront évalués par un schéma de turbulence KPP (Large *et al.*, 1994). Les coefficients de turbulence atmosphérique (K_s^a et K_m^a) seront quant à eux évalués avec un schéma TKE (Cuxart *et al.*, 2000). Il s'agit de deux schémas de turbulence classiquement utilisés dans des modèles opérationnels tel que le modèle atmosphérique Arpège de Météo-France pour le schéma TKE, ou les modèles océaniques Hycom et Croco pour le schéma KPP.

6.1.4 Récapitulatif

Notre modèle simple colonne est donc une simplification des équations primitives 3D auxquelles nous avons ajouté plusieurs hypothèses :

Hypothèses supplémentaires du modèle simple colonne

- Homogénéité horizontale : $\frac{\partial}{\partial x} = \frac{\partial}{\partial y} = 0$
- Advection verticale négligée : $\langle w \rangle \approx 0$
- Atmosphère transparente : l'atmosphère n'absorbe pas les flux radiatifs

Équations modèle simple colonne

$$\begin{array}{ll}
 \frac{\partial \mathbf{u}_a}{\partial t} = -f_C \mathbf{k} \times (\mathbf{u}_a - \mathbf{u}_{aG}) + \frac{\partial}{\partial z} \left(K_m^a \frac{\partial \mathbf{u}_a}{\partial z} \right) & \frac{\partial \mathbf{u}_o}{\partial t} = -f_C \mathbf{k} \times \mathbf{u}_o + \frac{\partial}{\partial z} \left(K_m^o \frac{\partial \mathbf{u}_o}{\partial z} \right) & \text{sur } \Omega_\alpha \times [0, T] \\
 \frac{\partial \theta_a}{\partial t} = \frac{\partial}{\partial z} \left(K_s^a \frac{\partial \theta_a}{\partial z} \right) + \lambda_S (\theta_a - \theta_{aLS}) & \frac{\partial \theta_o}{\partial t} = \frac{\partial}{\partial z} \left(K_s^o \frac{\partial \theta_o}{\partial z} + \frac{Q_S}{\rho_o C_p^o} \right) & \text{sur } \Omega_\alpha \times [0, T] \\
 \frac{\partial q_a}{\partial t} = \frac{\partial}{\partial z} \left(K_s^a \frac{\partial q_a}{\partial z} \right) + \lambda_S (q_a - q_{aLS}) & \frac{\partial S_o}{\partial t} = \frac{\partial}{\partial z} \left(K_s^o \frac{\partial S_o}{\partial z} \right) & \text{sur } \Omega_\alpha \times [0, T] \\
 K_m^a \frac{\partial \mathbf{u}_a}{\partial z} = \frac{\boldsymbol{\tau}}{\rho_a} & K_m^o \frac{\partial \mathbf{u}_o}{\partial z} = \frac{\boldsymbol{\tau}}{\rho_o} & \text{sur } \Gamma \times [0, T] \\
 K_s^a \frac{\partial \theta_a}{\partial z} = \frac{Q_H}{\rho_a C_p^a} & K_s^o \frac{\partial \theta_o}{\partial z} = \frac{Q_{\text{net}}}{\rho_o C_p^o} - \frac{Q_{S_0}}{\rho_o C_p^o} & \text{sur } \Gamma \times [0, T] \\
 K_s^a \frac{\partial q_a}{\partial z} = -\frac{E}{\rho_a} & K_s^o \frac{\partial S_o}{\partial z} = 0.001 (E - P) S_o(z^-) & \text{sur } \Gamma \times [0, T] \\
 \frac{\partial \mathbf{u}_a}{\partial z} = 0 & \frac{\partial \mathbf{u}_o}{\partial z} = 0 & \text{sur } \partial \Omega_\alpha^{\text{ext}} \times [0, T] \\
 \frac{\partial \theta_a}{\partial z} = 0 & \frac{\partial \theta_o}{\partial z} = v_s \cdot \frac{\partial \theta_o(z, 0)}{\partial z} \Big|_{z=-h_o} & \text{sur } \partial \Omega_\alpha^{\text{ext}} \times [0, T] \\
 \frac{\partial q_a}{\partial z} = 0 & \frac{\partial S_o}{\partial z} = v_s \cdot \frac{\partial S_o(z, 0)}{\partial z} \Big|_{z=-h_o} & \text{sur } \partial \Omega_\alpha^{\text{ext}} \times [0, T]
 \end{array} \tag{6.2}$$

Modèle atmosphérique
Modèle océanique

Dans ce modèle :

- $K_s^{a,o}$ et $K_m^{a,o}$ sont évalués grâce aux schémas de turbulence TKE (Cuxart *et al.*, 2000) et KPP (Large *et al.*, 1994),
- les différents flux sont évalués à partir des modèles et des formulations bulk,
- le couplage océan-atmosphère se fait à l'interface via la continuité des flux turbulents.

Pour conclure, nous avons défini entièrement le modèle simple colonne en présentant ses équations ainsi que les conditions aux frontières extérieures et à l'interface. Nous avons également défini les schémas de turbulence utilisés pour le calcul des différents coefficients de viscosité et de diffusion turbulente. Il s'agit d'un modèle simplifié par rapport aux équations primitives 3D complètes, mais permettant d'étudier deux problématiques majeures des modèles opérationnels : le calcul des flux d'interface via les formulations bulk et l'évaluation des

coefficients de turbulence dans la couche limite de surface via des paramétrisations complexes. Cela introduit dans le modèle des non-linéarités et non-différentiabilités qui peuvent s'avérer complexes à gérer dans un processus d'assimilation de données. Ce modèle devrait ainsi nous permettre de tester la robustesse de nos algorithmes d'assimilation de données en situation plus réaliste, tout comme l'ont fait Smith *et al.* (2015) avec une version colonne du modèle de prévision du CEPMMT : IFS¹. Nous pouvons cependant regretter l'absence de description précise de leur configuration pour réaliser un cas-test de référence.

6.2 Choix d'implémentation du modèle

Afin de compléter la description du modèle, il nous faut décrire plusieurs choix d'implémentation. Ces choix concernent d'une part la discrétisation spatiale et temporelle du modèle, et d'autre part la définition des différents flux qui ne sont pas évalués directement par le modèle.

6.2.1 Discrétisation temporelle

Nous discrétiserons les équations du modèle avec un schéma d'Euler implicite en temps, excepté pour le terme de Coriolis dans l'équation de quantité de mouvement atmosphérique, pour lequel nous utiliserons un schéma dit *forward-backward*. Ce schéma consiste à discrétiser alternativement de manière explicite puis implicite les équations par rapport aux composantes u_α et v_α . Le paramètre de Coriolis sera $f_C = 10^{-4} \text{ s}^{-1}$ et nous nous placerons sur une fenêtre temporelle de 7 jours en considérant la conformité en temps entre les deux modèles ($\Delta t_a = \Delta t_o = \Delta t$) :

- fenêtre temporelle : $T = 7 \text{ j}$,
- pas de temps : $\Delta t = 30 \text{ s}$.

6.2.2 Définition de la grille de calcul

Les domaines atmosphériques et océaniques sont définis par $\Omega_a = [0, h_a]$ et $\Omega_o = [-h_o, 0]$. Leurs grilles verticales sont évaluées en fonction de quatre paramètres :

- h_α la hauteur ou profondeur du modèle,
- $N_{L\alpha}$ le nombre de niveaux verticaux,
- $\theta_{s\alpha}$ un coefficient de *stretching* de la résolution : plus $\theta_{s\alpha}$ est élevé, plus la grille est raffinée proche de la surface Γ . Un maillage plus fin est utilisé en surface afin de mieux pouvoir représenter la dynamique du modèle qui y est beaucoup plus turbulente que dans l'atmosphère libre ou l'océan profond.
- $h_{c\alpha}$ la hauteur/profondeur de transition entre une grille quasi-uniforme (pour $z < h_{c\alpha}$ et $z > h_{c\alpha}$) et une grille non-uniforme (pour $z > h_{c\alpha}$ et $z < h_{c\alpha}$).

1. *Integrated Forecast System*. Pour plus d'informations sur ce modèle, voir <https://www.ecmwf.int/en/research/modelling-and-prediction/atmospheric-dynamics>

A partir de ces quatre paramètres nous pouvons définir l'altitude des centres des mailles z_k ainsi que l'altitude des interfaces entre les mailles $z_{k+\frac{1}{2}}$ telles que :

$$\begin{aligned} \text{(Centres)} \quad z_k &= h_{c\alpha} \kappa_k^\alpha + (h_\alpha - h_{c\alpha}) \frac{\sinh \kappa_k^\alpha \theta_{s\alpha}}{\sinh \theta_{s\alpha}} && \text{avec } k \in \llbracket 1, N_{La} \rrbracket \\ \text{(Interfaces)} \quad z_{k+\frac{1}{2}} &= h_{c\alpha} \kappa_{k+\frac{1}{2}}^\alpha + (h_\alpha - h_{c\alpha}) \frac{\sinh \kappa_{k+\frac{1}{2}}^\alpha \theta_{s\alpha}}{\sinh \theta_{s\alpha}} && \text{avec } k \in \llbracket 0, N_{La} \rrbracket \end{aligned}$$

où on définit $\kappa_k^a = \frac{k-\frac{1}{2}}{N_{La}}$ et $\kappa_k^o = \frac{k-\frac{1}{2}-N_{Lo}}{N_{Lo}}$. Dans la suite, nous prendrons les paramètres des tableaux 6.1 et 6.2 pour définir les grilles des modèles d'atmosphère et d'océan. La figure 6.1 représente la hauteur des interfaces des mailles sur tout le domaine océan-atmosphère. On observe un maillage très resserré proche de l'interface qui s'agrandit au fur et à mesure que l'on s'en éloigne. Pour avoir quelques ordres de grandeur, la taille des mailles proche de l'interface est de l'ordre de 1 m dans l'océan et de 20 m dans l'atmosphère. Le premier niveau océanique est donc à environ 50 cm de profondeur et le premier niveau atmosphérique à 10 m d'altitude.

Description	Coefficient
Nombre de niveaux	$N_{La} = 51$
Toit du modèle	$h_a = 2000 \text{ m}$
Altitude de transition	$h_{c\alpha} = 200 \text{ m}$
Coefficient de stretching	$\theta_{s\alpha} = 2$

Tableau 6.1 – Coefficients de la grille du modèle colonne atmosphérique

Description	Coefficient
Nombre de niveaux	$N_{Lo} = 50$
Profondeur du modèle	$h_o = 500 \text{ m}$
Profondeur de transition	$h_{co} = 50 \text{ m}$
Coefficient de stretching	$\theta_{so} = 6.5$

Tableau 6.2 – Coefficients de la grille du modèle colonne océanique

6.2.3 Flux atmosphériques imposés

Il nous faut ensuite définir les différents flux imposés au modèle, à savoir le flux radiatif solaire Q_S , le flux radiatif Q_I et le flux de précipitations P . Nous définissons ces flux sur la fenêtre temporelle $[-3, 7]$ jours. Nous expliquerons dans la suite pourquoi nous prenons en compte ces trois jours en amont la fenêtre temporelle de 7 jours $[0, T]$.

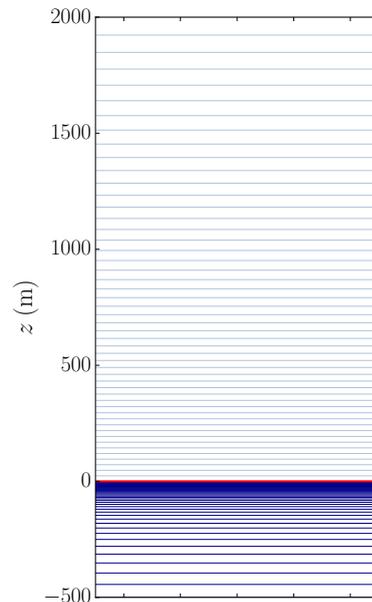


FIGURE 6.1 – Représentation graphique de la grille du modèle couplé. La hauteur des interfaces des mailles atmosphériques (respectivement océaniques) sont représentées en bleu clair (resp. bleu foncé). L'interface Γ océan-atmosphère est représentée par une ligne rouge à l'altitude $z = 0$.

Flux solaire

Le flux solaire $Q_S(0, t) = Q_{S_0}$ arrivant en surface de l'océan est défini de telle sorte à simuler un cycle diurne du soleil avec une variabilité d'amplitude sur la fenêtre temporelle :

$$Q_{S_0} = \begin{cases} \mathcal{Q}(t) \cdot Q_{S_{\max}} \cdot \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) & \forall t \in [21600 + l \cdot J, 64800 + l \cdot J] \text{ s} \\ 0 & \text{avec } J = 86400 \text{ s et } l \in \llbracket -3, 6 \rrbracket \\ & \text{sinon} \end{cases}$$

$$\text{avec } \mathcal{Q}(t) = 0.8 + 0.2 \cdot \cos\left(\frac{t}{40000}\right)$$

$$Q_{S_{\max}} = 420 \text{ W.m}^{-2}$$

$$\sigma = 7200 \text{ s}$$

$$\mu = 43200 + l \cdot J \text{ s}$$

Le flux solaire est donc nul entre 18 h et 6 h, quand il fait nuit, et suit la forme d'une gaussienne le jour entre 6 h et 18 h. La hauteur de la gaussienne est modulée par $\mathcal{Q}(t)$ qui permet d'ajouter une variabilité du flux sur la fenêtre temporelle, permettant par exemple de représenter l'effet des nuages. La figure 6.2a représente $Q_{S_0}(t)$.

Flux de précipitations

En ce qui concerne le flux $P(t)$, nous imposerons un épisode de fortes précipitations peu avant le troisième jour de la fenêtre temporelle :

$$P(t) = 9 \cdot \exp\left(-\frac{(t-\mu)^2}{2\sigma^2}\right) \text{ cm.jour}^{-1} \quad \forall t \in [-3\text{j}, 7\text{j}]$$

avec $\mu = 200000 \text{ s}$
 $\sigma = 30000 \text{ s}$

Le flux de précipitation $P(t)$ est représenté dans la figure 6.2b.

Remarque 6.2.1

Nous avons ici écrit le flux de précipitations en cm.jour^{-1} pour que cela soit plus explicite. Cependant, ce flux devant être exprimé en $\text{kg.m}^{-2}.\text{s}^{-1}$, il nous faudra le convertir sachant que $1 \text{ kg.m}^{-2}.\text{s}^{-1}$ est équivalent à $8640 \text{ cm.jour}^{-1}$ d'eau.

Flux radiatif à grandes longueurs d'onde

Enfin, nous choisissons un flux radiatif $Q_{L\downarrow}$ constant sur la fenêtre temporelle :

$$Q_{L\downarrow}(t) = 350 \text{ W.m}^{-2} \quad \forall t \in [-3\text{j}, 7\text{j}]$$

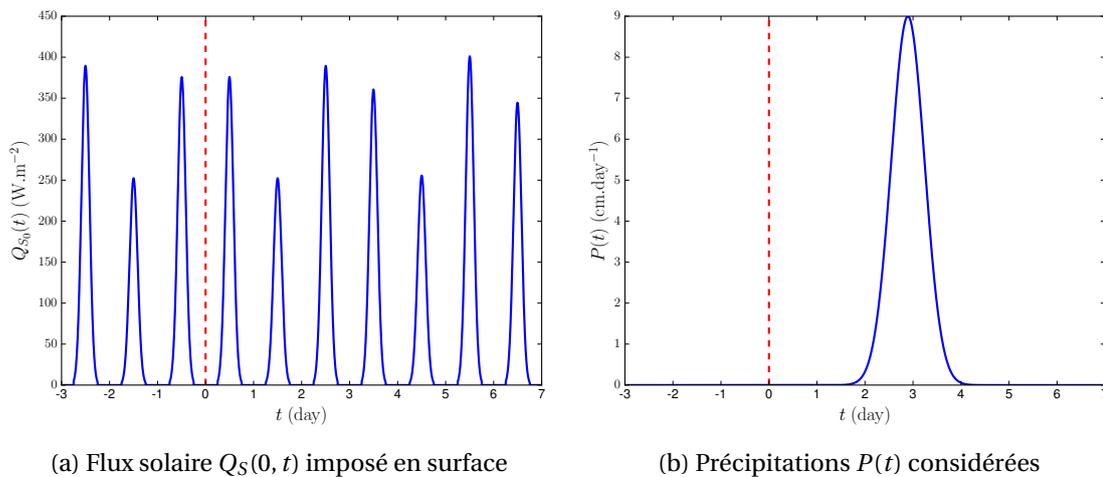


FIGURE 6.2 – Représentation du flux solaire en surface de l'océan $Q_S(0, t)$ (a) et des précipitations $P(t)$ (b) imposées au modèle simple colonne.

6.2.4 Rappels grande échelle

Pour finaliser la description du modèle, il nous reste à définir les rappels grande échelle \mathbf{u}_{ag} , θ_{aLS} et q_{aLS} . Nous considérerons simplement que :

$$\begin{aligned} u_{aG}(z, t) &= \begin{cases} u_{a0}(z) + U_{up} \cdot \mathcal{G}_{\mathbf{u}}(z, t) & \text{si } z \leq h_{\text{lim}} \\ u_{a0}(z) & \text{si } z > h_{\text{lim}} \end{cases} & z \in \Omega_a, t \in [0, T] \\ v_{aG}(z, t) &= \begin{cases} v_{a0}(z) + V_{up} \cdot \mathcal{G}_{\mathbf{u}}(z, t) & \text{si } z \leq h_{\text{lim}} \\ v_{a0}(z) & \text{si } z > h_{\text{lim}} \end{cases} & z \in \Omega_a, t \in [0, T] \\ \theta_{aLS}(z, t) &= \theta_{a0}(z) + \Theta_{up} \cdot \mathcal{G}_{\theta}(z, t) & z \in \Omega_a, t \in [0, T] \\ q_{aLS}(z, t) &= q_{a0}(z) & z \in \Omega_a, t \in [0, T] \end{aligned} \quad (6.3)$$

où u_{a0} , v_{a0} , θ_{a0} et q_{a0} sont les conditions initiales du modèle. Les constantes U_{up} , V_{up} , Θ_{up} ainsi que les fonctions \mathcal{G} seront définies au cas par cas dans le chapitre 8 lors de nos différentes expériences. Nous ferons attention à assurer la continuité des champs en $z = h_{\text{lim}}$.

6.3 Stratégies de couplage

Le modèle étant entièrement décrit, il nous reste à définir une ou plusieurs stratégies de couplage. Pour cela considérons les vecteurs d'états atmosphériques et océaniques suivants :

$$\mathbf{x}_a = \begin{pmatrix} u_a \\ v_a \\ \theta_a \\ q_a \end{pmatrix}, \quad \mathbf{x}_o = \begin{pmatrix} u_o \\ v_o \\ \theta_o \\ S_o \end{pmatrix},$$

et nous pouvons alors ré-écrire les équations du modèle (6.2) de manière synthétique :

$$\left\{ \begin{array}{ll} \mathcal{M}_a(\mathbf{x}_a) = f_a & \text{sur } \Omega_a \times [0, T] \\ \mathbf{x}_a(z, 0) = \mathbf{x}_{a,0} & \text{sur } \Omega_a \\ \rho_a K_x^a \frac{\partial \mathbf{x}_a}{\partial z} = \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}, \mathbf{x}_{o|\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \\ \frac{\partial \mathbf{x}_a}{\partial z} = 0 & \text{sur } \partial\Omega_a^{\text{ext}} \times [0, T] \end{array} \right\} \left\{ \begin{array}{ll} \mathcal{M}_o(\mathbf{x}_o) = f_o & \text{sur } \Omega_o \times [0, T] \\ \mathbf{x}_o(z, 0) = \mathbf{x}_{o,0} & \text{sur } \Omega_o \\ \rho_o K_x^o \frac{\partial \mathbf{x}_o}{\partial z} = \mathcal{F}_{ao}(\mathbf{x}_{a|\Gamma}, \mathbf{x}_{o|\Gamma}, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \\ \frac{\partial \mathbf{x}_o}{\partial z} = \mathbf{v} \odot \frac{\partial \mathbf{x}_{o,0}}{\partial z} \Big|_{z=-h_o} & \text{sur } \partial\Omega_o^{\text{ext}} \times [0, T] \end{array} \right. \quad (6.4)$$

avec \odot le produit de Schur et $\mathbf{v} = (0, 0, \nu_s, \nu_s)^T$. Les opérateurs \mathcal{F}_{oa} et \mathcal{F}_{ao} cachent les formulations bulk permettant de calculer les différents flux à l'interface air-mer. Enfin, l'état initial du modèle est désigné par $\mathbf{x}_0 = (\mathbf{x}_{a,0}, \mathbf{x}_{o,0})^T$. Le problème du couplage est alors de trouver une solution du modèle (6.4). Nous envisagerons dans la suite deux stratégies de couplage par fenêtres temporelles, un couplage asynchrone et un couplage par itérations de Schwarz. Comme précédemment dans le chapitre 5, nous pouvons écrire ces deux algorithmes de couplage sous un formalisme identique.

Algorithmes de couplage

Nous décomposons la fenêtre $[0, T]$ en N_j fenêtres temporelles $T_j = [t_j, t_{j+1}]$ telles que $[0, T] = \bigcup_{j=0}^{N_j-1} T_j$. Nous proposons de trouver une solution du modèle (6.4) par itérations suc-

cessives telles que, pour $k > 0$:

$$\left\{ \begin{array}{ll} \mathcal{M}_a(\mathbf{x}_a^k) = f_a & \text{sur } \Omega_a \times T_j \\ \mathbf{x}_a^k(z, t_j) = \mathbf{x}_{a,j} & \text{sur } \Omega_a \\ \rho_a K_x^a \frac{\partial \mathbf{x}_a^k}{\partial z} = \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^{k-1}, \mathcal{R}) & \text{sur } \Gamma \times T_j \\ \frac{\partial \mathbf{x}_a^k}{\partial z} = 0 & \text{sur } \partial\Omega_a^{\text{ext}} \times T_j \end{array} \right. \left\{ \begin{array}{ll} \mathcal{M}_o(\mathbf{x}_o^k) = f_o & \text{sur } \Omega_o \times T_j \\ \mathbf{x}_o^k(z, t_j) = \mathbf{x}_{o,j} & \text{sur } \Omega_o \\ \rho_o K_x^o \frac{\partial \mathbf{x}_o^k}{\partial z} = \mathcal{F}_{ao}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^k, \mathcal{R}) & \text{sur } \Gamma \times T_j \\ \frac{\partial \mathbf{x}_o^k}{\partial z} = \mathbf{v} \odot \frac{\partial \mathbf{x}_{o,0}}{\partial z} \Big|_{z=-h_o} & \text{sur } \partial\Omega_o^{\text{ext}} \times T_j \end{array} \right.$$

où k dénote les itérations du modèle et $\mathbf{x}_j = (\mathbf{x}_{a,j}, \mathbf{x}_{o,j})^T$ est l'état initial sur la fenêtre temporelle T_j . Lorsque $j > 0$, cet état initial est fourni par la solution du modèle sur la fenêtre temporelle précédente. Le choix du first-guess $\mathbf{x}_{o|\Gamma}^0$ est différent selon si l'on est sur la première fenêtre temporelle ou non :

- si $j = 0$, $\mathbf{x}_{o|\Gamma}^0$ est choisi de manière arbitraire (bruit blanc, constante, estimation...)
- si $j > 0$, $\mathbf{x}_{o|\Gamma}^0$ est la solution à l'interface sur la fenêtre temporelle précédente T_{j-1} .

En écrivant le modèle ainsi, nous pouvons considérer un couplage asynchrone ou un couplage itératif de Schwarz par fenêtres temporelles, en fonction du nombre d'itérations k du modèle :

- si nous limitons les itérations à $k_{\max} = 1$, nous n'effectuons qu'une intégration du modèle sur chaque sous-fenêtre, ce qui correspond à un couplage **asynchrone**,
- si nous allons à convergence, c'est à dire que $\| \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^{k-1}, \mathcal{R}) - \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^k, \mathcal{R}) \| < \varepsilon^c$, avec ε^c un critère de convergence, alors nous serons dans le cas d'un couplage par algorithme de Schwarz.

Dans les deux cas, le modèle couplé reçoit en entrée d'intégration le vecteur d'état $(\mathbf{x}_o, \mathbf{x}_o^0)^T \in \Omega \times (\Gamma \times T_0)$. Nous noterons $\mathbf{x}^{k_{\max}}(t) = \mathcal{M}_{0,t}^{k_{\max}}(\mathbf{x}_o, \mathbf{x}_{o|\Gamma}^0)$ l'intégration du modèle jusqu'au temps t lorsque l'on stoppe l'algorithme de Schwarz à k_{\max} itération, et $\mathbf{x}^{\text{cvg}}(t) = \mathcal{M}_{0,t}^{\text{cvg}}(\mathbf{x}_o, \mathbf{x}_{o|\Gamma}^0)$ lorsque nous allons à convergence.

Modèles découplés

Certains algorithmes d'assimilation de données considérant deux modèles découplés d'océan et d'atmosphère, nous souhaitons ici également présenter leurs équations. En utilisant le même formalisme que précédemment, nous pouvons écrire ces deux modèles ainsi :

$$\left\{ \begin{array}{ll} \mathcal{M}_a(\mathbf{x}_a) = f_a & \text{sur } \Omega_a \times [0, T] \\ \mathbf{x}_a(z, t_j) = \mathbf{x}_{a,0} & \text{sur } \Omega_a \\ \rho_a K_x^a \frac{\partial \mathbf{x}_a}{\partial z} = \mathcal{F}_{oa}(\mathbf{x}_{a|\Gamma}^0, \mathbf{x}_{o|\Gamma}^0, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \\ \frac{\partial \mathbf{x}_a}{\partial z} = 0 & \text{sur } \partial\Omega_a^{\text{ext}} \times [0, T] \end{array} \right. \left\{ \begin{array}{ll} \mathcal{M}_o(\mathbf{x}_o) = f_o & \text{sur } \Omega_o \times [0, T] \\ \mathbf{x}_o(z, t_j) = \mathbf{x}_{o,0} & \text{sur } \Omega_o \\ \rho_o K_x^o \frac{\partial \mathbf{x}_o}{\partial z} = \mathcal{F}_{ao}(\mathbf{x}_{a|\Gamma}^0, \mathbf{x}_{o|\Gamma}^0, \mathcal{R}) & \text{sur } \Gamma \times [0, T] \\ \frac{\partial \mathbf{x}_o}{\partial z} = \mathbf{v} \odot \frac{\partial \mathbf{x}_{o,0}}{\partial z} \Big|_{z=-h_o} & \text{sur } \partial\Omega_o^{\text{ext}} \times [0, T] \end{array} \right.$$

où $\mathbf{x}_\Gamma^0 = (\mathbf{x}_{a|\Gamma}^0, \mathbf{x}_{o|\Gamma}^0)^T$ est la condition d'interface initiale du modèle. Le modèle découplé d'océan et d'atmosphère reçoit donc en entrée d'intégration le vecteur d'état $(\mathbf{x}_o, \mathbf{x}_\Gamma^0)^T \in \Omega \times (\Gamma \times [0, T])^2$. Nous noterons $\mathbf{x}^{\text{unc}}(t) = \mathcal{M}_{0,t}^{\text{unc}}(\mathbf{x}_o, \mathbf{x}_\Gamma^0)$ l'intégration du modèle découplé jusqu'au temps t .

Récapitulatif

Nous venons de décrire dans son ensemble un modèle couplé simple colonne non-linéaire avec paramétrisations sous-maille. Nous avons tout d'abord défini les équations du modèle avant d'explicitier les conditions au limites qui nous ont permis d'introduire le couplage entre les deux modèles. Nous avons ensuite explicité différents choix d'implémentation numérique, concernant la discrétisation des équations, les flux imposés au modèle ou encore la définition des termes de rappel grande échelle. Enfin, nous avons présenté deux stratégies de couplage que nous implémenterons ainsi que la formulation des deux modèles découplés d'océan et d'atmosphère. Nous allons à présent implémenter numériquement ce modèle afin d'y assimiler des données. Nous avons réalisé cette implémentation en Fortran et C++ au sein de OOPS, une surcouche logicielle développée au CEPMMT que nous allons introduire dans le chapitre suivant.

Chapitre 7

Implémentation de modèles et d'algorithmes dans OOPS

Sommaire

7.1 Introduction à OOPS	120
7.1.1 Principe de fonctionnement	120
7.1.2 Motivations à l'utilisation de OOPS	121
7.2 Implémentation du modèle couplé simple colonne dans OOPS	121
7.2.1 Architecture de OOPS	121
7.2.2 Fichiers de configuration	122
7.2.3 Réalisation d'une prévision et assimilation couplées	123
7.3 Implémentation du terme J^s dans OOPS	127
7.3.1 Structure des objets représentant les termes de la fonction coût	127
7.3.2 Création de <code>CostJs</code> , objet représentant le terme J^s	129
7.4 Test du gradient et évaluation de la fonction coût linéaire	129

Ce chapitre, plus technique que le reste du manuscrit, porte sur l'implémentation numérique dans OOPS¹ (*Object-Oriented Prediction System*) du modèle couplé simple colonne présenté dans le chapitre précédent et de nos algorithmes d'assimilation de données couplée (chapitre 4). OOPS est une surcouche logicielle développée au CEPMMT dédiée à la prévision et à l'assimilation de données, permettant de gérer plus aisément une diversité de systèmes complexes. La première partie de ce chapitre introduit cette surcouche logicielle et décrit son fonctionnement global. Puis nous irons un peu plus loin dans les détails techniques en présentant l'implémentation numérique du modèle couplé et de ses composantes découplées, ainsi que la réalisation d'une prévision. Enfin, nous présenterons le principe de fonctionnement de l'assimilation de données au sein de OOPS. Nous analyserons alors comment reproduire dans OOPS les différentes méthodes d'assimilation couplée introduites dans le chapitre 4. Nous verrons cependant que certains algorithmes ne peuvent pas être reproduits. Nous détaillerons alors les développements que cela a nécessité dans OOPS afin d'y parvenir.

1. Voir la présentation de Trémolet (2015) par exemple

7.1 Introduction à OOPS

Comme nous avons pu le décrire, un système de prévision complet regroupe de nombreux modèles : atmosphère, surface continentale, océan, glace, aérosols, etc. Tous ces modèles deviennent de plus en plus complexes et il est alors difficile de les faire interagir entre eux, notamment dans le but de les coupler. De même, chaque modèle possède souvent son propre système d'assimilation de données, rendant complexe le développement d'algorithmes d'assimilation couplée. De nombreuses solutions existent aujourd'hui pour gérer et faire interagir cette disparité de systèmes, comme par exemple OpenPALM (Fouilloux et Piacentini, 1999), Verdandi (Chapelle *et al.*, 2013), OpenDA², DART³, PDAF⁴, etc. Nous allons nous pencher sur une solution proposée par le CEPMMT pour répondre à ce besoin de modularité entre les modèles et les méthodes d'assimilation de données, **OOPS**.

7.1.1 Principe de fonctionnement

OOPS est une surcouche logicielle développée en C++ et Fortran dont le principe fondamental est de séparer les modèles de leurs systèmes d'assimilation, de prévision, etc. C'est à dire, par exemple, qu'un modèle peut utiliser n'importe quel algorithme d'assimilation implémenté dans OOPS. L'intérêt du langage C++ est la possibilité de créer des objets, afin de rendre le tout très modulaire. Schématiquement, les modèles ne sont pas implémentés dans OOPS, mais simplement « connectés » à ce dernier. Cette connexion est réalisée via les **interfaces** de OOPS. Nous pouvons alors exécuter une prévision ou une assimilation de données avec un modèle connecté grâce aux **applications** implémentées dans OOPS.

Les **modèles** sont indépendants de OOPS. Ils doivent cependant disposer d'un certain nombre d'objets compréhensibles par OOPS, permettant leur utilisation avec les différentes applications. En dehors de ces objets, la programmation du modèle est laissée totalement libre à l'utilisateur.

Les objets du modèle compréhensibles par OOPS sont les **interfaces**. Il s'agit d'objets qui permettent de « connecter » le modèle à OOPS.

Les **applications** sont des objets exécutables de OOPS. Une application peut être exécutée avec n'importe quel modèle correctement connecté.

Ce principe de fonctionnement est récapitulé par la figure 7.1. Il est donc possible de considérer OOPS comme une « boîte noire », et ne connaître que son modèle pour effectuer diverses expériences d'assimilation, de prévision, etc. Nous analyserons dans la suite comment « connecter » un modèle à OOPS.

2. <http://openda.org>

3. <http://www.image.ucar.edu/DAReS/DART>

4. <http://pdaf.awi.de>

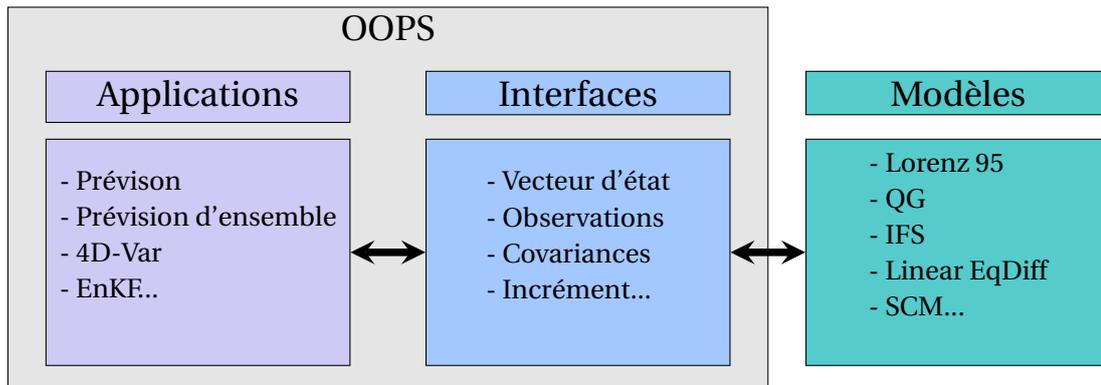


FIGURE 7.1 – Principe de fonctionnement de OOPS : les applications sont exécutées avec un modèle, compréhensible par OOPS grâce aux interfaces qui font le lien entre les deux. Linear EqDiff fait référence à notre modèle de diffusion linéaire introduit dans le chapitre 5, et SCM au modèle couplé non-linéaire simple colonne introduit dans le chapitre précédent.

7.1.2 Motivations à l'utilisation de OOPS

Nous souhaitons évaluer la performance des différents algorithmes d'assimilation de données introduits dans le chapitre 4 avec le modèle non-linéaire simple colonne introduit précédemment (chapitre 6). Assimiler des données sur un tel modèle requiert plus de puissance de calcul qu'avec notre précédent modèle de diffusion (chapitre 5). Il peut alors être intéressant d'optimiser numériquement notre code. Le C++ et le Fortran offrent une belle opportunité en ce sens, grâce à la programmation objet et une meilleure gestion de la mémoire que dans d'autres langages, tel que le Python. Il est donc intéressant d'intégrer notre système à OOPS, afin de profiter de ses performances numériques mais surtout des différentes applications qui y sont déjà implémentées et optimisées (4D-Var incrémental avec une liste de minimiseurs). De plus, l'intégration d'un tel modèle dit « jouet » à OOPS pourrait être bénéfique à la communauté d'assimilation de données. En effet, dans le dernier rapport de l'Organisation Météorologique Mondiale concernant l'assimilation de données couplée, il a été mentionné le besoin pour la recherche d'avoir à disposition des modèles jouets au sein de systèmes tel que OOPS (Penny *et al.*, 2017). Enfin, nous pouvons noter que Météo-France est en train d'adapter son système d'assimilation opérationnel pour l'intégrer à OOPS et l'utiliser par la suite. Il nous semble donc intéressant de prendre en main ce logiciel pour y implémenter nos algorithmes et notre modèle simple colonne non-linéaire. Cela permettrait à ce modèle d'être réutilisé par exemple dans d'autres centres de recherche, et à nos algorithmes d'être testés sur des modèles opérationnels. Cela nous permet aussi de partager ce que l'on a pu faire dans OOPS au projet ERA-Clim2, dans lequel s'inscrit cette thèse.

7.2 Implémentation du modèle couplé simple colonne dans OOPS

7.2.1 Architecture de OOPS

Afin d'implémenter notre modèle colonne dans OOPS, il nous faut tout d'abord en comprendre l'architecture afin de « connecter » correctement le modèle. Comme nous l'avons mentionné, un modèle est connecté à OOPS via l'intermédiaire d'un vingtaine de classes d'interfaces, telles que :

- le vecteur d'état,
- les équations du modèle,
- les types d'observations, etc.

Chaque interface possède également un certain nombre de méthodes à implémenter. Le tableau A.1 disponible en annexe A récapitule l'ensemble des 20 classes d'interfaces ainsi que leurs méthodes inhérentes à implémenter dans un modèle pour pouvoir le connecter correctement à OOPS. Ces interfaces sont en fait des patrons (ou *templates*) de classe. Il s'agit d'une spécificité du langage C++ permettant de créer des classes dites *génériques*. Celles-ci servent alors de patron sur lesquelles nous viendrons calquer différentes classes de notre modèle. L'interfaçage des modèles est réalisé grâce aux *alias*, qui permettent de lier une classe du modèle à une classe d'interface. Les interfaces sont ensuite utilisées par OOPS pour réaliser l'ensemble des applications. En effet, toutes les applications sont codées en fonction des classes d'interfaces, les rendant ainsi indépendantes des modèles. Un modèle correctement connecté à OOPS via les interfaces pourra donc utiliser les différentes applications implémentées. Inversement, une application créée en fonction des classes d'interfaces pourra être utilisée par n'importe quel modèle connecté.

Remarque 7.2.1 - Taille de OOPS

OOPS est un code d'une taille raisonnable, puisque l'ensemble des applications, interfaces et classes associées représente environ 28 000 lignes de code. En revanche, OOPS utilise, entre autre, la bibliothèque `eckit` développée au CEPMMT. Cette librairie représente environ 92 000 lignes de code, et permet par exemple la gestion des fichiers de configuration pour exécuter les différentes applications. La connaissance précise de cette librairie n'est pas requise, mais certaines bases comme les fichiers de configurations sont nécessaires à maîtriser.

7.2.2 Fichiers de configuration

Pour exécuter les différentes applications de OOPS, il est nécessaire de fournir à ce dernier un fichier de configuration permettant de définir, par exemple, la configuration du modèle utilisé, la date, le temps d'intégration du modèle, le type d'algorithme d'assimilation, les sorties du modèle, etc. Toutes ces informations sont structurées dans un fichier JSON⁵. Il s'agit d'un format d'échange de données facilement compréhensible. Le code 7.1 nous donne un exemple de fichier de configuration pour réaliser une prévision avec un modèle donné. Dans ce fichier, nous définissons les différents types de messages qui seront affichés dans le terminal dans la rubrique `logging`. Les rubriques `resolution` et `model` définissent les différents paramètres du modèle. `initial` définit l'état initial fourni au modèle, `forecast_length` le temps d'intégration du modèle et `output` différentes sorties du modèle. Notons que certains paramètres du fichier de configurations sont inhérents au modèle utilisé, tel que `resolution`. Les champs et le type de données à fournir dans ces catégories dépendent donc directement du modèle utilisé. D'autres paramètres, comme `forecast_length`, sont destinés à OOPS, et doivent donc être du format demandé par OOPS.

5. *JavaScript Object Notation*, voir <https://www.json.org/> pour plus d'informations.

Remarque 7.2.2 - La gestion du temps dans OOPS

Les différents temps et durées dans OOPS sont définis à partir du formalisme suivant : PdDThHmMsS où d, h, m et s définissent respectivement le nombre de jours, d'heures, de minutes et de secondes de la durée considérée. Nous remarquerons également que si d = 0, alors nous n'écrirons pas 0D dans la notation (de même pour les autres unités de temps).

Code 7.1 – Exemple d'un fichier de configuration JSON

```

1 {
2   "logging": {
3     "categories": "Info, Test, Warning, Error"
4   },
5   "resolution": {
6     "resol": "40"
7   },
8   "model": {
9     "tstep": "PT1H30M",
10    "f": "8.0"
11  },
12  "initial": {
13    "date": "2010-01-01T00:00:00Z",
14    "filename": "Data/test.an.2010-01-01T00:00:00Z.195"
15  },
16  "forecast_length": "P3D",
17  "output": {
18    "frequency": "PT1H30M",
19    "datadir": "Data",
20    "exp": "test",
21    "type": "fc",
22    "date": "2010-01-01T00:00:00Z"
23  }
24 }

```

7.2.3 Réalisation d'une prévision et assimilation couplées**Stratégies d'implémentation**

A titre d'illustration, nous allons nous intéresser à la réalisation d'une prévision avec notre modèle couplé introduit au chapitre précédent. Nous allons présenter ici les différents choix que nous avons fait pour réaliser cette prévision au sein de OOPS. En suivant le principe de séparation des applications et des modèles, nous pouvons envisager deux possibilités d'implémentation d'un modèle couplé (voir figure 7.2) :

- (a) le couplage est réalisé par OOPS auquel nous fournissons deux modèles,
- (b) le couplage est géré par le modèle connecté à OOPS.

OOPS ne disposant pas d'applications du type *Prévision couplée*, nous écarterons la proposition (a). En effet, celle-ci nécessitera le développement de nouvelles applications, et très certainement de nouvelles interfaces, ce qui n'est pas souhaitable. Il nous semble également com-

plexe d'arriver à gérer le couplage de manière abstraite, permettant de satisfaire l'ensemble des modèles et des stratégies de couplage. Nous considérerons alors l'option (b) pour implémenter notre modèle couplé. En pratique, nous avons développé séparément les deux modèles d'atmosphère et d'océan que nous avons connectés à OOPS. Le modèle couplé utilise alors ces deux modèles et gère le couplage entre eux. Lors d'une prévision couplée, les deux modèles d'océan et d'atmosphère sont alors invisibles à OOPS qui ne communique qu'avec le modèle couplé. Cette stratégie d'implémentation est récapitulée dans la figure 7.3.

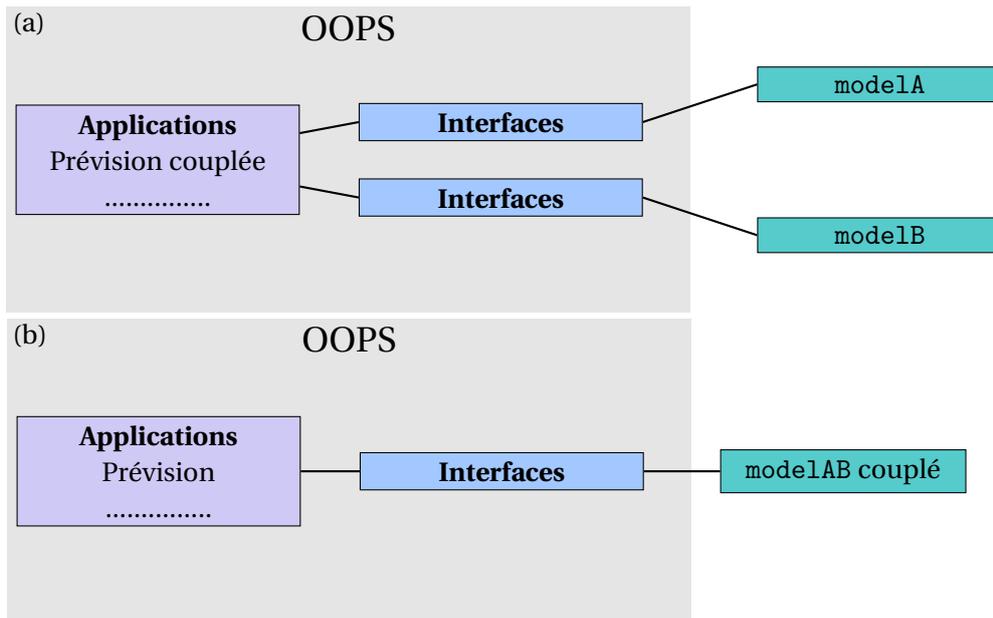


FIGURE 7.2 – Deux options de couplage pour deux modèles couplés dans OOPS. L'option (a) suppose que l'on puisse les connecter simultanément à OOPS qui gèrera ensuite la prévision couplée. L'option (b) considère que le couplage est géré au sein même du modèle, OOPS perçoit alors le modèle couplé comme un unique modèle et ne distingue pas les différentes composantes qui le compose.

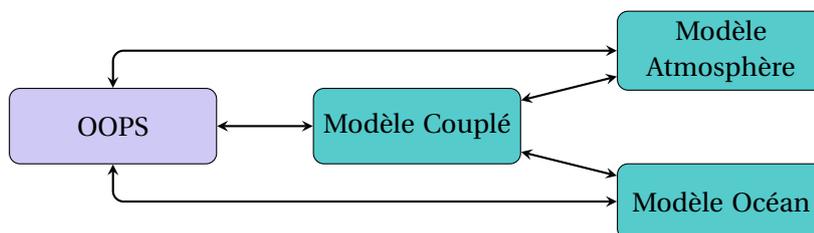


FIGURE 7.3 – Implémentation de nos modèles au sein de OOPS. Nous avons deux modèles d'océan et d'atmosphère indépendants l'un de l'autre connectés à OOPS, et un modèle couplé également connecté à OOPS, qui utilise et couple ces deux précédents modèles.

Réalisation de la prévision

Les prévisions sont réalisées par l'application `Forecast`. Cette application, après avoir instancié différents objets tels que la résolution du modèle, ses équations, ou encore son vecteur

d'état, exécute la routine `forecast` de l'interface `Model` (interface qui gère l'évolution du vecteur d'état et du modèle à partir de ses équations). La routine `forecast` est décrite par le code 7.2.

Code 7.2 – Routine `forecast` de l'interface `Model`, permettant de réaliser une prévision

```

1  template<typename MODEL>
2  void Model<MODEL>::forecast(State_ & xx, const ModelAux_ & maux,
3                               const util::Duration & len,
4                               PostProcessor<State_> & post) const {
5      /* Routine forecast de la classe d'interface Model
6       * Reçoit en paramètre :
7       * - le vecteur d'état (interface State_)
8       * - un biais, forçages externes... (interface ModelAux_)
9       * - la durée du forecast (len)
10      * - un PostProcessor : objet qui gère par exemple les sorties du modèle
11      */
12      LOG(Trace) << "Model<MODEL>::forecast starting";
13
14      const util::DateTime end(xx.validTime() + len); //Calcul de la date de fin d'intégration
15      LOG(Info) << "Model:forecast: forecast starting: " << xx;
16      this->initialize(xx); //Initialise le vecteur d'état et le modèle
17      post.initialize(xx, end, model->timeResolution()); //Initialise le PostProcessor
18      while (xx.validTime() < end) { //Boucle jusqu'au dernier pas de temps
19          this->step(xx, maux); //Un pas de temps faisant évoluer le modèle et le vecteur d'état
20          post.process(xx); //Exécution du PostProcessor
21      }
22      post.finalize(xx); //Etape finale du PostProcessor
23      this->finalize(xx); //Etape finale de l'intégration du modèle
24      LOG(Info) << "Model:forecast: forecast finished: " << xx;
25      LOG(Trace) << "Model<MODEL>::forecast done";
26  }

```

Nous remarquons que cette routine est constituée d'une boucle `while` représentant l'ensemble des pas de temps du modèle. La routine `step`, correspondant à une intégration d'un pas de temps, fait évoluer le vecteur d'état et met à jour le temps auquel il se trouve. Arrivé à la fin d'intégration (la date du vecteur d'état correspondant à `end`) nous sortons de la boucle pour finaliser la prévision. Le couplage des modèles doit donc intervenir dans la routine `step`. L'algorithme 7.1 présente le principe de notre stratégie de couplage utilisée dans `step` afin de reproduire un couplage par itérations de Schwarz ou asynchrone. Le critère d'arrêt de notre algorithme est alors $(xx.validTime() == end \ \&\& \ ECI < \epsilon_C)$ où ECI représente l'erreur de couplage à l'interface entre deux itérations ($\mathcal{F}_{oa}(\mathbf{x}_{a|_r}^k, \mathbf{x}_{o|_r}^{k-1}, \mathcal{R}) - \mathcal{F}_{oa}(\mathbf{x}_{a|_r}^k, \mathbf{x}_{o|_r}^k, \mathcal{R})$) et ϵ_C est le critère de convergence que l'on se définit. Il est donc possible de réaliser une prévision couplée au sein de OOPS via un algorithme de Schwarz, en gérant le couplage au sein du modèle et en s'arrangeant pour que OOPS n'achève pas la prévision prématurément. Nous n'avons donc pas à implémenter de nouvelles applications dans OOPS permettant de réaliser une prévision couplée.

Enfin, nous introduisons ici les `PostProcessor` (notés PP dans la suite). Ce sont des objets servant à gérer différents processus, et sont eux-même composés de plusieurs objets `PostBase` (notés PB dans la suite). A chaque exécution du PP (`post.process(x)`; par exemple dans le code 7.2), tous les PB sont exécutés un à un et réalisent diverses actions comme l'affichage de messages à l'écran, l'écriture dans un fichier de l'état du modèle, la sauvegarde de la trajectoire du modèle etc.

Algorithme 7.1 Principe de la routine `step` du modèle couplé

Entrée : $\mathbf{x} = (\mathbf{x}_{a,i}, \mathbf{x}_{o,j})^T$	<i>Vecteurs d'état aux temps t_i et t_j</i>
if <code>modelTurn == atm</code> then	<i>Si nous devons intégrer le modèle atmosphérique</i>
$\mathbf{x}_{a,i+1} = M_{a,i,i+1}(\mathbf{x}_{a,i})$	<i>Pas de temps du modèle atmosphérique</i>
if $t_{i+1} = t_N$ then	<i>Si nous arrivons à la fin d'intégration du modèle</i>
$D_o(\mathbf{x}_o) = F_{ao}(\mathbf{x}_a, \mathbf{x}_o, \mathcal{R})$	<i>Couplage atmosphère → océan</i>
$j = 0$	<i>On réinitialise le temps de l'océan au début de la fenêtre</i>
<code>modelTurn = oce</code>	<i>Mise à jour de la variable <code>modelTurn</code></i>
end if	
else	<i>Sinon, nous devons intégrer le modèle océanique</i>
$\mathbf{x}_{o,j+1} = M_{o,j,j+1}(\mathbf{x}_{o,j})$	<i>Pas de temps du modèle océanique</i>
if $t_{j+1} = t_N$ && $ECl \geq \text{epsilonC}$ then	<i>Fin d'intégration sans avoir convergé</i>
$D_a(\mathbf{x}_a) = F_{oa}(\mathbf{x}_a, \mathbf{x}_o, \mathcal{R})$	<i>Couplage océan → atmosphère</i>
$i = 0$	<i>On réinitialise le temps de l'atmosphère au début de la fenêtre</i>
<code>modelTurn = atm</code>	<i>Mise à jour de la variable <code>modelTurn</code></i>
end if	
end if	

Assimilation de données avec le modèle couplé

Nous nous intéressons à présent à assimiler des données au sein du modèle couplé. L'application `Variational` propose d'assimiler des données avec un 4D-Var incrémental au sein d'un modèle. Cette application nécessite de recevoir diverses informations provenant du fichier de configuration permettant de définir, entre autres :

- la fonction coût à minimiser (matrices de covariance, observations, ébauche),
- la configuration du modèle direct,
- le choix du minimiseur,
- la configuration des boucles externes incluant :
 - ◇ la configuration des modèles linéaires et adjoints,
 - ◇ le critère de convergence de la minimisation et un nombre maximal d'itérations.

Si le modèle a correctement été implémenté et connecté à OOPS, et que l'ensemble des informations nécessaires a été fourni, OOPS gère alors l'assimilation de données. Notons que la définition de la matrice \mathbf{B} ou encore la gestion des observations font partie du modèle et doivent être correctement implémentées et interfacées avec OOPS. Enfin, la gestion des modèles linéaires et adjoints est similaire à celle du modèle direct. Nous pouvons donc utiliser la même stratégie que précédemment pour implémenter les modèles linéaires et adjoints couplés.

Remarque 7.2.3 - Dérivation des modèles linéaires et adjoints

Dans notre cas, nous avons différencié nos modèles d'atmosphère et d'océan avec le différenciateur automatique `Tapenade` développé à l'INRIA (Hascoët et Pascual, 2013). Puis nous avons dérivé manuellement le modèle couplé, en utilisant les modèles linéaires et adjoints d'océan et d'atmosphère.

Remarque 7.2.4 - Taille des modèles

Le code du modèle atmosphérique représente environ 34 000 lignes, celui du modèle océanique 24 000 lignes et le code du modèle couplé utilisant ces deux composantes environ 10 000 lignes. Il s'agit donc de modèles de taille raisonnable. En comparaison, un modèle de prévision opérationnel d'atmosphère comme Arpège à Météo-France fait environ 1.5 million de lignes. Un modèle d'océan comme NEMO est plus petit et représente environ 200 000 lignes.

Finalement, l'assimilation de données nécessite « seulement » d'implémenter quelques interfaces supplémentaires par rapport à la prévision, comme par exemple Covariance, ObsSpace, ObsVector (voir le tableau A.1).

Définition de la fonction coût

Il est possible de définir chaque terme de la fonction coût J . Cela est réalisé à partir du fichier de configuration utilisé pour exécuter l'application Variational. Les différents termes de J sont des objets C++, et sont gérés par la classe de OOPS CostFunction. A l'heure actuelle, trois types de termes sont implémentés dans OOPS :

- différentes formes de J^b , pour quantifier l'écart à l'ébauche,
- différentes formes de J^o , pour quantifier les écarts aux observations,
- J^{dfi} , un terme de filtrage (*digital filter initialization*).

J^{dfi} est un terme permettant de filtrer des signaux haute fréquence qui peuvent se créer lors de l'assimilation de données (création d'ondes de gravité par exemple). Ces signaux dégradent la qualité de l'analyse et impactent directement les prévisions à court terme. Il existe plusieurs méthodes basées sur ce filtre, dont le filtre Dolph-Chebyshev qui est implémenté dans OOPS (voir par exemple Lynch *et al.* (1999) pour plus de détails sur ce filtre). Pour conclure, il sera relativement aisé d'assimiler des données sur notre modèle couplé avec un algorithme type MH-C ou MI-C. En revanche, pour les méthodes présentant un terme de contrainte faible de couplage au sein de la fonction coût, il n'existe pas d'équivalent de J^S implémenté dans OOPS. Nous allons donc expliquer comment nous avons pu l'implémenter.

7.3 Implémentation du terme J^S dans OOPS

Rappelons tout d'abord la forme du terme J^S (4.1) :

$$\begin{aligned} J^S(\mathbf{x}_0) &= \frac{1}{2} \sum_{i=0}^N \left(D_a(\mathbf{x}_{a|\Gamma,i}^{k_{\max}}) - D_o(\mathbf{x}_{o|\Gamma,i}^{k_{\max}}) \right)^T \mathbf{W}^{-1} \left(D_a(\mathbf{x}_{a|\Gamma,i}^{k_{\max}}) - D_o(\mathbf{x}_{o|\Gamma,i}^{k_{\max}}) \right) \\ &= \frac{1}{2} \sum_{i=0}^N \left\| \left(D_a(\mathbf{x}_{a|\Gamma,i}^{k_{\max}}) - D_o(\mathbf{x}_{o|\Gamma,i}^{k_{\max}}) \right) \right\|_{\mathbf{W}^{-1}}^2 \end{aligned}$$

Nous commencerons par présenter comment sont implémentés les différents termes de la fonction coût avant de décrire comment nous avons pu implémenter un nouvel objet CostJs dans OOPS, représentant le terme J^S .

7.3.1 Structure des objets représentant les termes de la fonction coût

Pour comprendre comment construire ce nouvel objet CostJs, intéressons nous à la structure des termes de la fonction coût au sein de OOPS.

Fonction coût

Considérons $J = J^b + J^o + J^{dfi}$ la fonction coût à minimiser par OOPS. Chaque terme de J est un objet de OOPS, et tous ces objets sont indépendants les uns des autres. OOPS gère simplement l'évaluation de la fonction coût et sa minimisation en manipulant ces objets, sans savoir explicitement ce qui s'y trouve. Ces objets dérivent des types :

- `CostJbTotal` pour J^b
- `CostTermBase` pour les autres termes (J^o , J^{dfi} ...), que l'on notera CTB.

Les CTB sont les objets génériques des termes de la fonction coût, excepté pour J^b qui est à part. Ils doivent contenir un certain nombre de méthodes, telles que `multiplyCovar` ou `multiplyCoInv` gérant la multiplication par la matrice de covariance ou son inverse (par exemple multiplication par \mathbf{R} ou \mathbf{R}^{-1}).

Remarque 7.3.1 - À propos de l'implémentation de J^b

Le terme J^b est représenté différemment car son rôle au sein de la fonction coût est différent de celui des autres termes. En effet, J^b devra gérer les différentes interactions avec l'ébauche et la multiplication par la matrice \mathbf{B} , permettant le pré-conditionnement de certains algorithmes de minimisation.

Boucle externe

On évalue la valeur de la fonction coût et on effectue différents diagnostics nécessaires à la minimisation (comme les écarts aux observations) au sein de la boucle externe. Pour réaliser cela, OOPS demande à J^b et à chaque CTB la création d'un objet PB qui lui est lié. OOPS fournit alors ce PB à l'intégration du modèle non-linéaire. Au cours de la prévision, il est possible aux PB de réaliser une action à chaque pas de temps (par exemple, sauvegarder la trajectoire pour la comparer ensuite aux observations). A la fin de l'intégration, chaque CTB (et J^b) réalisent leurs diagnostics à partir de leurs PB respectifs (évaluer les écarts aux observations par exemple). Les PB devront donc avoir enregistré toutes les informations nécessaires au cours de l'intégration. OOPS dispose alors de l'évaluation de la fonction coût en boucle externe. Le terme d'ébauche ainsi que les CTB sont alors prêts pour la minimisation en boucle interne.

Boucle interne

La minimisation dans la boucle interne est une succession d'intégrations du modèle linéaire et adjoint. De même que pour l'intégration du modèle non-linéaire, chaque CTB (et J^b) crée un objet avant chaque intégration afin de pouvoir réaliser la minimisation :

- un objet de type `PostBaseTL` (PBTL) pour l'intégration du modèle linéaire, permettant l'évaluation de la fonction coût linéaire,
- un objet de type `PostBaseAD` (PBAD) pour l'intégration du modèle adjoint, permettant de mettre à jour le modèle adjoint en fonction des informations fournies par l'intégration du modèle linéaire.

Remarque 7.3.2

Les objets `CostTermBase`, `PostBase`, `PostBaseTL` et `PostBaseAD` sont des classes abstraites de OOPS. Les différents types de termes de la fonction coût sont donc des objets dérivant de CTB. Ils fourniront à OOPS des objets dérivant de PB, PBTL et PBAD pour l'intégration des modèles.

7.3.2 Création de `CostJs`, objet représentant le terme J^s

Maintenant que nous avons une meilleure compréhension du fonctionnement des termes de la fonction coût, nous allons présenter la classe `CostJs` permettant de représenter le terme J^s . Comme tout CTB, `CostJs` doit pouvoir créer des objets PB, PBTl et PBAD permettant d'évaluer la fonction coût et d'appliquer une correction à la variable adjointe durant la minimisation. Nous avons alors créé les classes `InterfDiff`, `InterfDiffTL` et `InterfDiffAD` dérivant de ces objets. Elles permettent de calculer l'erreur de couplage à l'interface durant l'intégration du modèle direct et tangent, et d'appliquer sa contribution adjointe durant l'intégration du modèle reverse. `CostJs` dispose également d'un objet `ModelSpaceCovarianceBase`, permettant de représenter la matrice \mathbf{W} . La définition de \mathbf{W} se fera via le fichier de configuration JSON, dans la rubrique définissant les paramètres de `CostJs`. Une fois ces développements terminés, nous pouvons alors assimiler des données avec notre modèle couplé et un terme J^s présent dans la fonction coût. Il nous est donc possible de réaliser l'ensemble des algorithmes d'assimilation couplée que nous avons pu décrire dans le chapitre 4.

7.4 Test du gradient et évaluation de la fonction coût linéaire

L'implémentation numérique d'un modèle, de sa linéarisation et de son adjoint n'est pas quelque chose d'évident. L'implémentation d'un opérateur d'observation non plus. De ce fait, pour vérifier le bon fonctionnement de toutes les composantes d'un système d'assimilation, il est possible de réaliser des tests numériques. OOPS propose toute une batterie de tests unitaires permettant de vérifier chaque composante du système indépendamment, comme l'approximation linéaire du modèle ou le modèle adjoint. Cependant, un test qui nous semble essentiel manque à cette bibliothèque : le test du gradient. Ce test du gradient permet de vérifier si le calcul du gradient ∇J , associé à la fonction coût J , est réalisé correctement. Le principe du test est de vérifier que :

$$\lim_{\alpha \rightarrow 0} \frac{J(\delta \mathbf{x} + \alpha \mathbf{B} \nabla J(\delta \mathbf{x})) - J(\delta \mathbf{x})}{\alpha \|\mathbf{B} \nabla J(\delta \mathbf{x})\|^2} = 1 \quad (7.1)$$

où $\delta \mathbf{x}$ est l'incrément de la fonction coût du 4D-Var incrémental. Ce test vient du développement de Taylor suivant :

$$J(\delta \mathbf{x} + \alpha \mathbf{B} \nabla J(\delta \mathbf{x})) = J(\delta \mathbf{x}) + \alpha \|\mathbf{B} \nabla J(\delta \mathbf{x})\|^2 + \mathcal{O}(\alpha \|\mathbf{B} \nabla J(\delta \mathbf{x})\|^2)$$

Remarque 7.4.1

Le test du gradient permet de tester plusieurs composantes du système en un seul test : modèle linéaire, adjoint et opérateur d'observation. Nous supposons que ce test n'a pas été implémenté du fait qu'il est possible de tester l'ensemble de ces composantes séparément avec les tests unitaires.

Pour pouvoir réaliser ce test du gradient (7.1), il nous faut accéder à la valeur de la fonction coût linéaire $J(\delta \mathbf{x})$. Or, OOPS ne permet pas d'y accéder directement. Nous avons donc implémenté une nouvelle routine au sein de la classe `CostFunction` permettant de calculer la fonction coût linéaire. Une fois le calcul de la fonction coût linéaire rendu possible, nous avons pu implémenter le test du gradient dans la bibliothèque de tests de OOPS.

Récapitulatif et retours d'expérience

OOPS est une surcouche logicielle développée en C++, visant à répondre au problème de diversité des modèles de prévisions jointe à la diversité des méthodes d'assimilation, de prévision etc. Le principe est que l'on puisse connecter n'importe quel modèle à OOPS afin de pouvoir réaliser les différentes applications qui y sont implémentées (assimilation de données, etc). Les modèles sont donc complètement indépendants de OOPS, et leur connexion est réalisée via une vingtaine de classes d'interfaces. Connecter un modèle à OOPS n'est cependant pas si évident, puisqu'il n'existe pas actuellement de documentation détaillée ou de guide concernant les interfaces. Nous avons donc du découvrir par nous mêmes le principe de chaque interface et les routines qui les composent. Nous proposons en annexe A un tableau répondant partiellement à ce problème de documentation (tableau A.1). Nous y décrivons succinctement l'ensemble des interfaces et la liste des méthodes à implémenter pour chacune d'entre elles. Nous n'avons malheureusement pas eu le temps de les documenter avec plus de détails. Enfin, nous avons implémenté dans OOPS la possibilité de rajouter le terme J^s dans la fonction coût. Cela nous a permis de mieux comprendre le fonctionnement interne de OOPS. Nous avons également ajouté la possibilité d'effectuer un test du gradient.

Pour conclure sur l'utilisation de OOPS, sa prise en main n'a pas été immédiate et a pris du temps. Cependant, nous avons pu valoriser cela avec l'implémentation de notre modèle couplé ainsi que le rajout du terme J^s pour l'assimilation de données. De plus, nous avons également pu ajouter le test du gradient dans la bibliothèque de tests de OOPS. Nous pouvons alors espérer que ce modèle, ainsi que la contrainte faible de couplage, soient ré-utilisés par la suite.

Chapitre 8

Évaluation numérique de quelques algorithmes d'assimilation couplée dans un cadre non-linéaire

Sommaire

8.1	Création d'un jeu de données	131
8.1.1	Création du cas test de référence	132
8.1.2	État vrai, ébauche et observations	138
8.2	Description des expériences numériques	141
8.2.1	Choix des observations et covariances d'erreur	142
8.2.2	Contrainte faible de couplage	143
8.2.3	Algorithmes considérés	144
8.3	Résultats et discussions	146
8.3.1	Après 10 boucles externes	147
8.3.2	Après 2 boucles externes	150

Ce huitième et dernier chapitre porte sur l'évaluation des algorithmes d'assimilation de données couplée présentés dans le chapitre 4, dans un cadre non-linéaire. Nous allons tester la robustesse de ces algorithmes grâce au modèle couplé simple colonne non-linéaire présenté dans le chapitre 6. Ce modèle a été implémenté au sein de OOPS, une surcouche logicielle développée au CEPMMT, que nous venons de présenter. Nous commencerons par créer un jeu de données afin de réaliser diverses expériences. Ce jeu de données sera constitué d'un état vrai, d'une ébauche et d'observations du système auquel on s'intéresse. Il sera produit à partir d'un cas test de référence que nous décrirons. Nous présenterons ensuite les différentes expériences envisagées ainsi que leurs résultats. Nous pourrions enfin conclure sur la robustesse des différents algorithmes d'assimilation de données couplée présentés au cours du manuscrit.

8.1 Création d'un jeu de données

Afin de définir un jeu de données pour réaliser diverses expériences d'assimilation, nous commençons par créer un cas test de référence du modèle couplé. Nous pourrions ensuite,

à partir de ce cas test, définir un état vrai, une ébauche et des observations. Le cas test sera caractérisé par :

- la stratégie de couplage considérée,
- l'état initial du système,
- le first-guess à l'interface.

Dans le chapitre 5, nous avons pu définir un état initial équilibré du modèle linéaire de manière analytique. Il est en revanche plus complexe d'en définir un pour le modèle non-linéaire simple colonne que nous utilisons ici, de manière analytique. Une stratégie classique est alors de créer un état initial à partir d'une première prévision dite *spin-up*, qui produira un état équilibré avec les équations du modèle. Nous commencerons donc par réaliser ce spin-up, à partir duquel nous créerons le cas test de référence.

8.1.1 Création du cas test de référence

Stratégie de couplage

Pour décrire le cas test, il nous faut tout d'abord choisir une stratégie de couplage. Les différentes stratégies de couplages implémentées dans le modèle ont été présentées à la fin du chapitre 6. Nous choisissons pour le cas test un couplage par itérations de Schwarz avec comme critère de convergence $\varepsilon^c = 10^{-2}$ tel que :

$$\| \mathcal{F}_{oa}(\mathbf{x}_{a|_r}^k, \mathbf{x}_{o|_r}^{k-1}, \mathcal{R}) - \mathcal{F}_{oa}(\mathbf{x}_{a|_r}^k, \mathbf{x}_{o|_r}^k, \mathcal{R}) \| < \varepsilon^c$$

Cette stratégie de couplage nécessite de définir un premier itéré $\mathbf{x}_{o|_r}^0$ sur la première fenêtre temporelle d'intégration T_0 . Ce first-guess, que nous définirons dans la suite, se compose :

- des courants de surface (SSU/SSV) $\mathbf{u}_o^0(0, t)$ avec $t \in T_0$,
- de la température de surface (SST) $\theta_o^0(0, t)$ avec $t \in T_0$.

Création du spin-up

Comme nous l'avons mentionné, nous allons créer l'état initial du cas-test de référence à partir d'un spin-up que nous allons décrire ici. Ce dernier sera défini sur la fenêtre temporelle de 10 jours $[-3, T]$, avec $T = 7$ j. Notons que les différents flux atmosphériques imposés (flux radiatifs et précipitations) ont bien été définis sur cette fenêtre $[-3, T]$ (figure 6.2 du chapitre 6). La description du spin-up se décompose en trois étapes :

1. la définition des conditions initiales (incluant le first-guess),
2. la définition du rappel grande échelle,
3. l'intégration du modèle.

Dans la suite le vecteur d'état du spin-up sera noté $\mathbf{x}^{\text{su}} = (\mathbf{x}_a^{\text{su}}, \mathbf{x}_o^{\text{su}})^T \in \Omega$.

Définition des conditions initiales (CI). L'état initial du spin up est défini au temps $t = -3$ j par $\mathbf{x}_{-3}^{\text{su}} = (\mathbf{x}_{a,-3}^{\text{su}}, \mathbf{x}_{o,-3}^{\text{su}})^T \in \Omega$ tel que :

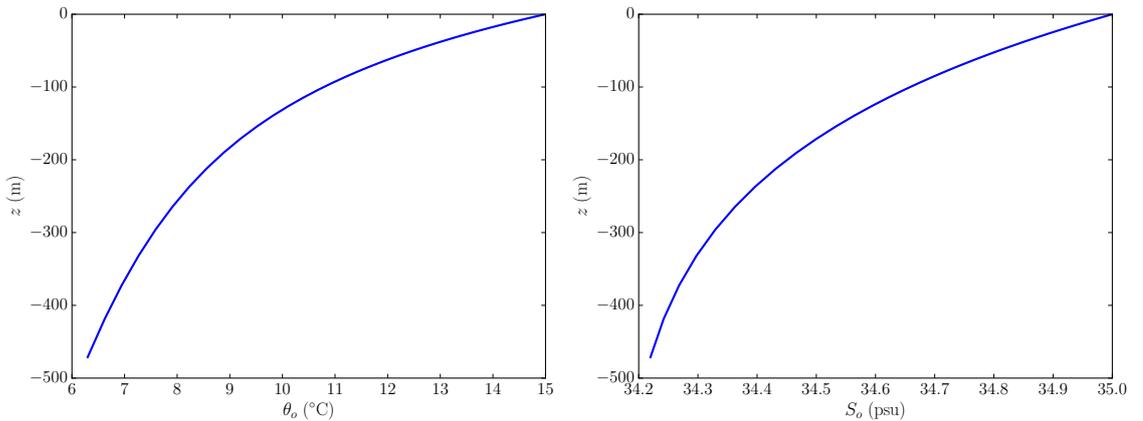
$$\begin{aligned} \mathbf{x}_{a,-3}^{\text{su}}(z) &= (u_{a,-3}^{\text{su}}(z), v_{a,-3}^{\text{su}}(z), \theta_{a,-3}^{\text{su}}(z), q_{a,-3}^{\text{su}}(z)), \\ \text{avec : } u_{a,-3}^{\text{su}}(z) &= 15 \text{ m} \cdot \text{s}^{-1} & z \in \Omega_a \\ v_{a,-3}^{\text{su}}(z) &= 3 \text{ m} \cdot \text{s}^{-1} & z \in \Omega_a \\ \theta_{a,-3}^{\text{su}}(z) &= \theta^{\text{ref}} + \frac{N^2 \theta^{\text{ref}}}{g} z \text{ } ^\circ\text{C} & z \in \Omega_a, \text{ avec } N = 0.01 \text{ s}^{-1} \text{ et } \theta^{\text{ref}} = 12.3 \text{ } ^\circ\text{C} \\ q_{a,-3}^{\text{su}}(z) &= 0.01 \text{ kg/kg} & z \in \Omega_a \end{aligned}$$

où N est la fréquence de Brunt-Väisälä et g l'accélération de la pesanteur. Côté océanique, nous choisissons les profils de CI suivants :

$$\begin{aligned} \mathbf{x}_{o,-3}^{\text{su}}(z) &= (u_{o,-3}^{\text{su}}(z), v_{o,-3}^{\text{su}}(z), \theta_{o,-3}^{\text{su}}(z), S_{o,-3}^{\text{su}}(z)), \\ \text{avec : } u_{o,-3}^{\text{su}}(z) &= 0 \text{ m} \cdot \text{s}^{-1} & z \in \Omega_o \\ v_{o,-3}^{\text{su}}(z) &= 0 \text{ m} \cdot \text{s}^{-1} & z \in \Omega_o \\ \theta_{o,-3}^{\text{su}}(z) &= 5 + 1.475 \cdot 10^{-3} \cdot z + 5 \cdot \exp\left(\frac{z}{100}\right) + 5 \cdot \exp\left(\frac{z}{500}\right) \text{ } ^\circ\text{C} & z \in \Omega_o \\ S_{o,-3}^{\text{su}}(z) &= 34 - 1.75 \cdot 10^{-4} \cdot z + 0.8 \exp\left(\frac{z}{200}\right) + 0.2 \cdot \exp\left(\frac{z}{400}\right) \text{ psu} & z \in \Omega_a \end{aligned}$$

La figure 8.1 représente $\theta_{o,-3}^{\text{su}}(z)$ et $S_{o,-3}^{\text{su}}(z)$. Enfin, le first-guess océanique pour la première itération de Schwarz sera :

$$\begin{aligned} \text{(SSU)} \quad u_{o|_r}^0(t) &= 0 & t \in \times [-3, T] \\ \text{(SSV)} \quad v_{o|_r}^0(t) &= 0 & t \in \times [-3, T] \\ \text{(SST)} \quad \theta_{o|_r}^0(t) &= 15 \text{ } ^\circ\text{C} & t \in \times [-3, T] \end{aligned}$$



(a) Profil de température initial $\theta_{o,-3}^{\text{su}}(z)$

(b) Profil de salinité initial $S_{o,-3}^{\text{su}}(z)$

FIGURE 8.1 – Représentation des profils initiaux de température (a) et de salinité (b) du spin-up dans l'océan.

Rappel grande échelle. Les équations du modèle atmosphérique présentent des termes de rappel géostrophique et grande échelle \mathbf{u}_{ag} , θ_{aLS} et q_{aLS} définis par (6.3). Pour le spin-up, nous considérerons un rappel vers la condition initiale.

Intégration du modèle. Le spin-up est ainsi entièrement défini et est intégré sur la fenêtre de 10 jours $[-3, T]$. Notons que l'algorithme de Schwarz a convergé en 7 itérations. Nous pouvons à présent créer le cas test de référence à partir de ce spin-up.

Création du cas test de référence

Définition des conditions initiales. Le cas test de référence est établi à partir de la solution du spin-up sur la fenêtre $[0, T]$. La condition initiale $\mathbf{x}_0 = (\mathbf{x}_{a,0}, \mathbf{x}_{o,0})^T \in \Omega$ est définie à partir de la solution du spin-up au temps $t = 0$, \mathbf{x}_0^{su} . L'état initial est alors en équilibre avec le modèle grâce l'intégration du spin-up sur la période $[-3, 0]$. Les figures 8.2 et 8.3 représentent les CI atmosphériques $\mathbf{x}_{a,0}$ et océaniques $\mathbf{x}_{o,0}$ respectivement. Enfin, le first-guess $\mathbf{x}_{o|_r}$ est déterminé à partir de la solution du spin-up à l'interface sur la fenêtre $[0, T]$ (figure 8.4).

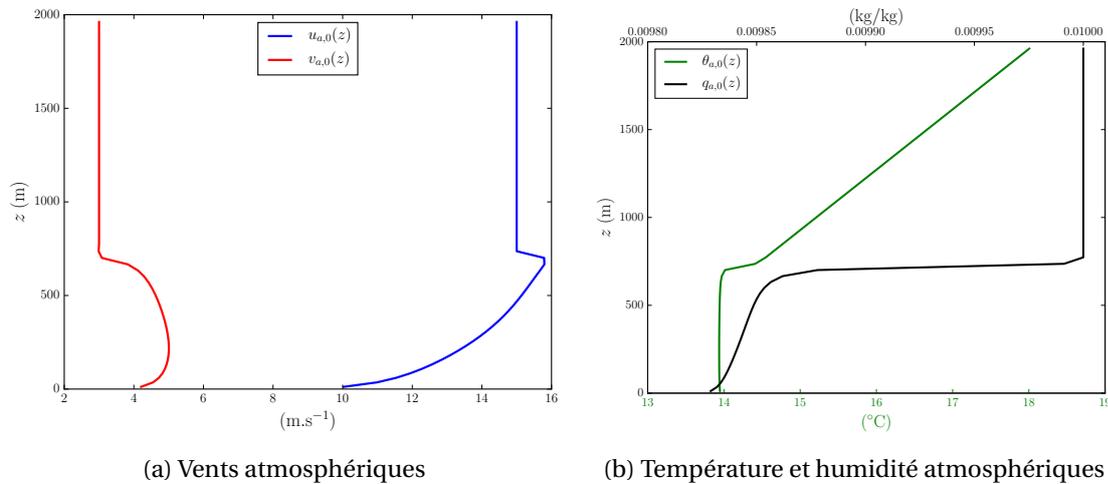


FIGURE 8.2 – Profils des conditions initiales atmosphériques du cas-test de référence. (a) représente les profils de vitesse des vents $u_{a,0}(z)$ et $v_{a,0}(z)$ et (b) les profils de température $\theta_{a,0}(z)$ et d'humidité $q_{a,0}(z)$.

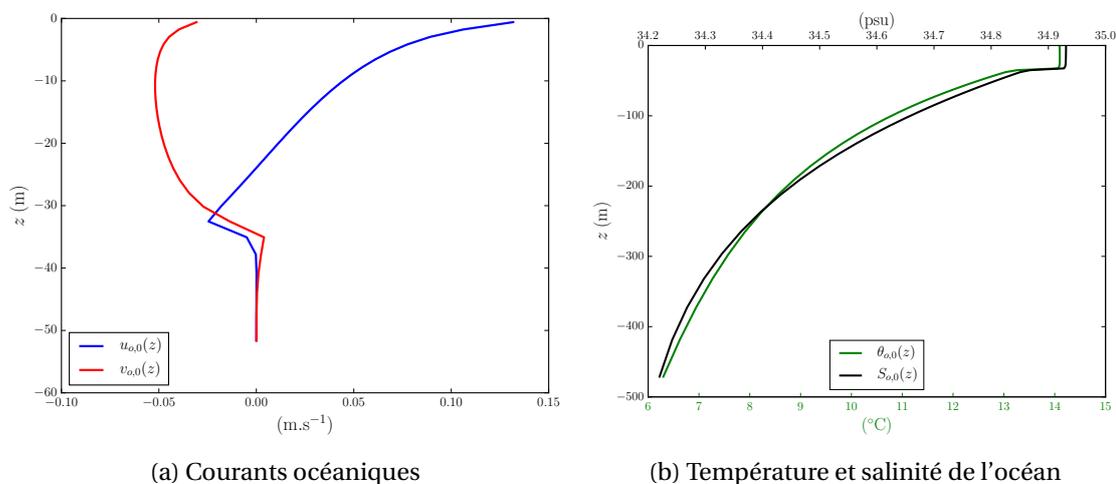


FIGURE 8.3 – Profils des conditions initiales océaniques du cas-test de référence. (a) représente les courants $u_{o,0}(z)$ et $v_{o,0}(z)$ ($u_o(z < 50, 0) = 0$ et $v_o(z < 50, 0) = 0$ ne sont pas représentés pour ne pas écraser l'échelle). (b) représente les profils de température $\theta_{o,0}(z)$ et de salinité $S_{o,0}(z)$.

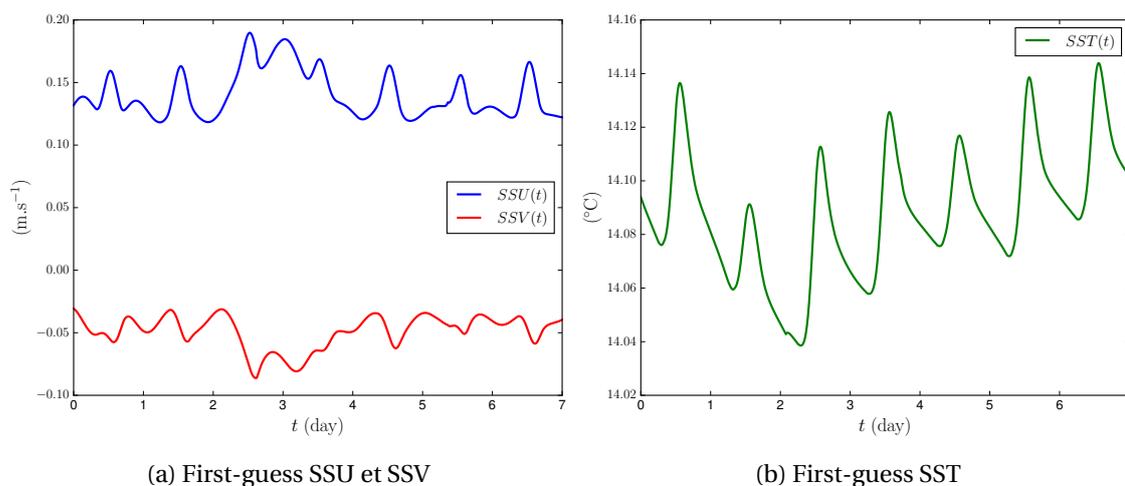


FIGURE 8.4 – First-guess océanique pour la SSU et SSV (a) et la SST (b).

Rappel grande échelle. Nous considérerons un rappel grande échelle différent dans le cas-test de référence que dans le spin-up. Le but est d'appliquer une variabilité des champs atmosphériques et océaniques permettant d'observer le passage de régimes stables à instables sur la fenêtre temporelle. Ce changement de régime nous semble important à observer puisqu'il sera intéressant de voir si les algorithmes d'assimilation couplée parviennent à les reproduire.

Remarque 8.1.1

On définit par **régime stable** le fait que $\theta_{a|_r} > \theta_{o|_r}$, l'air est alors plus dense à l'interface que sur les niveaux supérieurs. Inversement un **régime instable** est caractérisé par $\theta_{a|_r} < \theta_{o|_r}$, l'air est alors moins dense dans les basses couches. Dans cette situation, le système cherchera à rétablir un équilibre en créant des mouvements convectifs verticaux.

A partir des équations données en (6.3), nous choisissons les termes de rappel grande échelle suivants pour observer cette variabilité :

$$\begin{aligned}
 U_{up} &= 15 \text{ m}\cdot\text{s}^{-1} & \Theta_{up} &= -2.5 \text{ }^\circ\text{C} \\
 V_{up} &= 13 \text{ m}\cdot\text{s}^{-1} & \mathcal{G}_\theta(z, t) &= \cos\left(\frac{2\pi t}{T_p}\right) \cdot \left(\frac{Q_S(0, t)}{Q_{S_{\max}}} + 1\right) \cdot \exp\left(\frac{-15z}{h_a}\right) \\
 \mathcal{G}_u(z, t) &= \exp\left(-\frac{(t-\mu)^2}{2\cdot\sigma^2}\right) \cdot \sin\left(\frac{\pi z}{h_{\text{lim}}}\right)
 \end{aligned}$$

avec $\mu = 25 \cdot 10^4$ s, $\sigma = 6 \cdot 10^4$ s, $T_p = 86400$ s et $h_{\text{lim}} = 400$ m. Nous avons choisi d'appliquer une variation du rappel grande échelle θ_{LS} en fonction du flux solaire dans le but de renforcer le cycle diurne de température imposé par Q_S . Les figures 8.5 et 8.6 représentent ce forçage grande échelle du vent et de la température sur la fenêtre temporelle $[0, T]$.

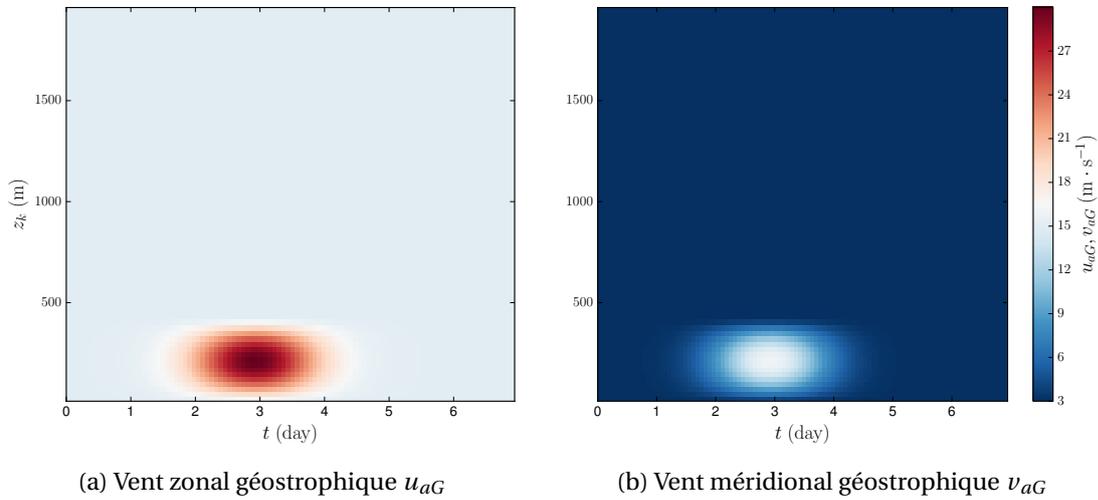


FIGURE 8.5 – Représentation du rappel grande échelle atmosphérique du vent. (a) représente le vent zonal u_{aG} et (b) le vent méridional v_{aG} . On observe une intensification des vents autour du troisième jour de la fenêtre temporelle.

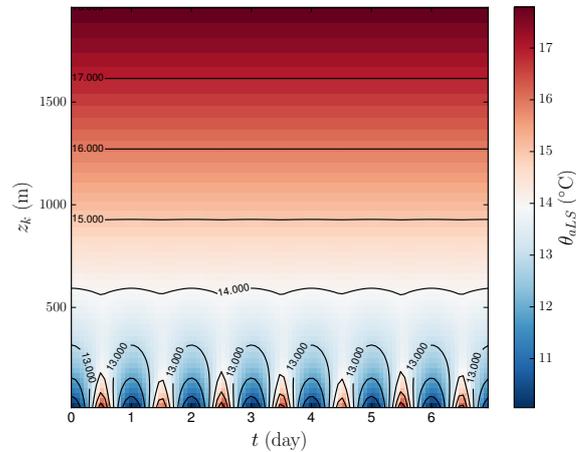


FIGURE 8.6 – Représentation du rappel grande échelle de la température potentielle atmosphérique θ_{aLS} . Nous pouvons observer le cycle diurne imposé par le flux solaire.

Intégration du modèle. Le modèle est à présent intégré sur la fenêtre temporelle $[0, T]$. L'algorithme de Schwarz converge en 6 itérations. La figure 8.7 représente l'évolution des vitesses de l'air et l'eau sur la fenêtre temporelle. Tout d'abord, nous pouvons remarquer une intensification de la vitesse du vent atmosphérique répondant à l'augmentation des vents géostrophiques. Cette augmentation a eu pour effet de rehausser le niveau de la couche limite atmosphérique autour de 1100 m avant de revenir à sa précédente altitude à la fin de l'épisode d'intensification des vents. Ensuite, nous pouvons observer des variations diurnes de la vitesse des courants zonaux, répondant au cycle diurne de Q_5 . De plus, une dynamique intéressante de l'océan est sa réponse à l'épisode de précipitations et d'intensification des vents. En effet, nous observons une augmentation de la vitesse des courants ainsi qu'une baisse de la couche limite durant cette période. Enfin, la figure (8.8) représente la différence de température à l'interface air-mer entre les deux milieux. Ce diagnostic nous permet d'observer le passage d'un régime instable à stable entre le deuxième et le troisième jour, avant de revenir à un régime instable au cours du quatrième jour de prévision. Nous obtenons donc bien la variabilité souhaitée pour notre cas-test de référence.

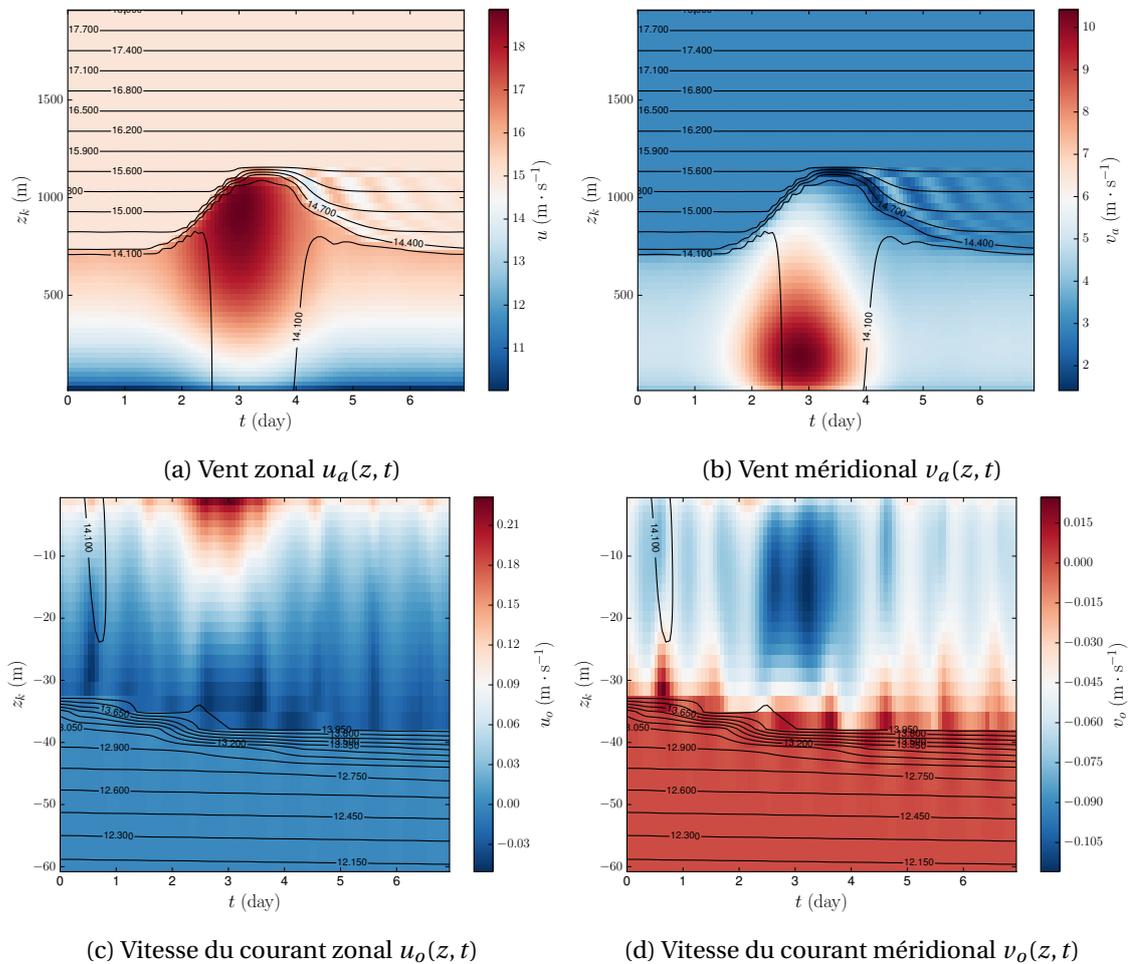


FIGURE 8.7 – Représentation des composantes zonale et méridienne du vent atmosphérique ((a) et (b)) et du courant océanique ((c) et (d)) définissant le cas-test de référence. Les iso-lignes noires décrivent le profil de température (en $^{\circ}\text{C}$).

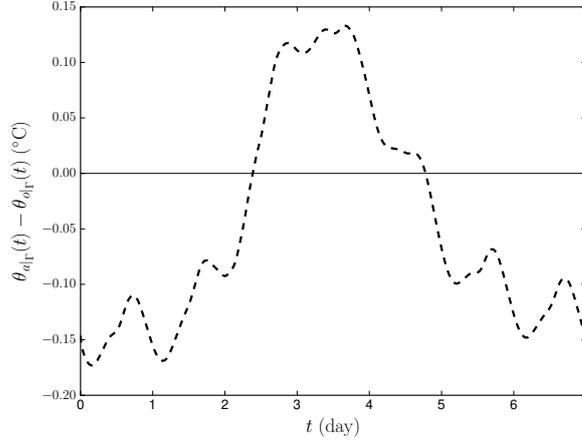


FIGURE 8.8 – Différence entre les températures de surface atmosphérique et océanique, permettant d’illustrer les changements de régimes sur la fenêtre temporelle (stable si $\theta_{a|_r} - \theta_{o|_r} > 0$, et instable si $\theta_{a|_r} - \theta_{o|_r} < 0$)

8.1.2 État vrai, ébauche et observations

Nous pouvons maintenant créer un état vrai, une ébauche et des observations, à partir du cas-test de référence complètement défini, pour les futures expériences d’assimilation de données.

État vrai

L’état-vrai est considéré comme étant exactement le cas-test de référence. Pour chaque expérience d’assimilation nous comparerons alors nos résultats à la solution du cas-test sur la fenêtre temporelle considérée. Nous notons $\mathbf{x}^t = (\mathbf{x}_a^t, \mathbf{x}_o^t)^T$ cet état vrai.

Ébauche

Nous définissons l’ébauche comme étant une prévision dont l’état initial présente une erreur par rapport à l’état vrai. Appliquer une erreur à l’état initial de référence risque cependant de le déséquilibrer par rapport aux équations du modèle. Il nous faut donc reprendre la stratégie du spin-up pour définir cette ébauche.

Spin-up de l’ébauche. Nous définissons les CI du spin-up de l’ébauche à partir des CI du spin-up de référence, auxquelles nous appliquons une erreur. Notons $\mathbf{x}_{-3}^{\text{su},b} = (\mathbf{x}_{a,-3}^{\text{su},b}, \mathbf{x}_{o,-3}^{\text{su},b})^T \in \Omega$ l’état initial du spin-up de l’ébauche tel que :

$$\mathbf{x}_{a,-3}^{\text{su},b}(z) = \left(u_{a,-3}^{\text{su},b}(z), v_{a,-3}^{\text{su},b}(z), \theta_{a,-3}^{\text{su},b}(z), q_{a,-3}^{\text{su},b}(z) \right),$$

$$\text{avec : } \begin{aligned} u_{a,-3}^{\text{su},b}(z) &= u_{a,-3}^{\text{su}}(z) + \tilde{u}_a^b(z) & z \in \Omega_a \\ v_{a,-3}^{\text{su},b}(z) &= v_{a,-3}^{\text{su}}(z) + \tilde{v}_a^b(z) & z \in \Omega_a \\ \theta_{a,-3}^{\text{su},b}(z) &= \theta_{a,-3}^{\text{su}}(z) + \tilde{\theta}_a^b(z) & z \in \Omega_a \\ q_{a,-3}^{\text{su},b}(z) &= q_{a,-3}^{\text{su}}(z) + \tilde{q}_a^b(z) & z \in \Omega_a \end{aligned}$$

et :

$$\mathbf{x}_{o,-3}^{\text{su},b}(z) = \left(u_{o,-3}^{\text{su},b}(z), v_{o,-3}^{\text{su},b}(z), \theta_{o,-3}^{\text{su},b}(z), S_{o,-3}^{\text{su},b}(z) \right),$$

$$\text{avec : } \begin{aligned} u_{o,-3}^{\text{su},b}(z) &= u_{o,-3}^{\text{su}}(z) + \tilde{u}_o^b(z) & z \in \Omega_o \\ v_{o,-3}^{\text{su},b}(z) &= v_{o,-3}^{\text{su}}(z) + \tilde{v}_o^b(z) & z \in \Omega_o \\ \theta_{o,-3}^{\text{su},b}(z) &= \theta_{o,-3}^{\text{su}}(z) + \tilde{\theta}_o^b(z) & z \in \Omega_o \\ S_{o,-3}^{\text{su},b}(z) &= S_{o,-3}^{\text{su}}(z) + \tilde{S}_o^b(z) & z \in \Omega_o \end{aligned}$$

où les différents champs \tilde{c}_α^b représentent les erreurs appliquées à une variable c . Nous considérons les erreurs suivantes pour définir le spin-up de l'ébauche :

$$\begin{aligned} \tilde{u}_a^b(z) &= -\frac{1}{2} (\tanh(z - 400) - 1) & \tilde{u}_o^b(z) &= 0 \text{ m.s}^{-1} \\ \tilde{v}_a^b(z) &= -\frac{0.3}{2} (\tanh(z - 400) - 1) & \tilde{v}_o^b(z) &= 0 \text{ m.s}^{-1} \\ \tilde{\theta}_a^b(z) &= 2.5 \text{ }^\circ\text{C} & \tilde{\theta}_o^b(z) &= -1.2 \text{ }^\circ\text{C} \\ \tilde{q}_a^b(z) &= 0 \text{ kg/kg} & \tilde{S}_o^b(z) &= 0.3 \text{ psu} \end{aligned}$$

Nous appliquons également l'erreur $\tilde{\theta}_o^b(0)$ au first-guess de la SST, et considérons le même first-guess que le spin-up de référence pour la SSU et SSV. Le spin-up de l'ébauche est alors intégré sur la fenêtre temporelle $[-3, T]$ via un algorithme itératif de Schwarz.

Création de l'ébauche. Nous définissons les conditions initiales et first-guess de l'ébauche à partir de la solution du spin-up de l'ébauche au temps $t = 0$, $\mathbf{x}_0^{\text{su},b}$. Nous intégrons alors le modèle avec un algorithme de Schwarz. La figure 8.9 présente les différences entre l'ébauche et l'état vrai pour les différents champs. Afin de juger de l'écart de l'ébauche par rapport à l'état vrai, il est possible de calculer l'erreur quadratique moyenne des différents champs sur la fenêtre d'intégration du modèle $[0, T]$:

$$\begin{aligned} \text{— } \text{rmse}_{u_a}^b &= 0.467 \text{ m.s}^{-1} & \text{— } \text{rmse}_{u_o}^b &= 3.63 \cdot 10^{-3} \text{ m.s}^{-1} \\ \text{— } \text{rmse}_{v_a}^b &= 0.457 \text{ m.s}^{-1} & \text{— } \text{rmse}_{v_o}^b &= 3.88 \cdot 10^{-3} \text{ m.s}^{-1} \\ \text{— } \text{rmse}_{\theta_a}^b &= 0.567 \text{ }^\circ\text{C} & \text{— } \text{rmse}_{\theta_o}^b &= 1.16 \text{ }^\circ\text{C} \\ \text{— } \text{rmse}_{q_a}^b &= 3.20 \cdot 10^{-4} \text{ kg/kg} & \text{— } \text{rmse}_{S_o}^b &= 0.301 \text{ psu} \end{aligned}$$

où rmse_c^b dénote l'erreur quadratique moyenne de la variable c^b par rapport à c^t . La figure 8.10 présente également l'évolution de la hauteur et profondeur des couches limites atmosphérique et océanique. Nous remarquons alors que les principales différences observées entre l'ébauche et l'état vrai se situent entre ces couches limites. Il sera donc intéressant d'observer si les algorithmes d'assimilation considérés parviennent à retrouver ces altitudes de couches limites. Enfin, nous pouvons observer dans quel régime se trouve l'ébauche. La figure 8.11 présente l'écart de température à la surface entre les deux milieux. Nous observons que l'ébauche se trouve dans un régime stable tout au long de la prévision. Il sera donc également intéressant d'observer si nous parvenons à changer de régime au cours de l'assimilation.

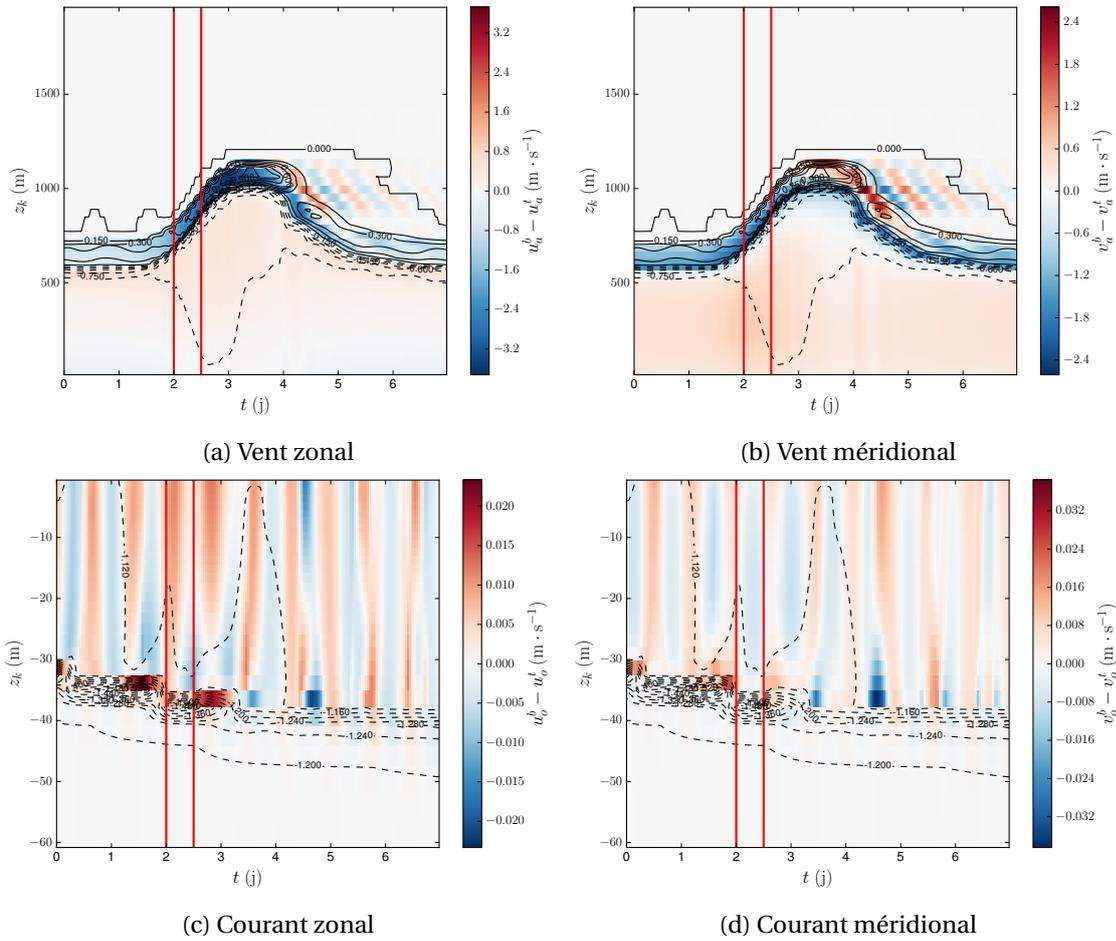


FIGURE 8.9 – Différences entre l'ébauche et l'état vrai. (a), (b), (c) et (d) représentent respectivement $u_a^b - u_a^t$, $v_a^b - v_a^t$, $u_o^b - u_o^t$ et $v_o^b - v_o^t$. Les isolignes correspondent aux écarts de température $\theta_a^b - \theta_a^t$ dans l'atmosphère, et $\theta_o^b - \theta_o^t$ dans l'océan. La période temporelle délimitée par les lignes rouges correspond à notre future fenêtre d'assimilation.

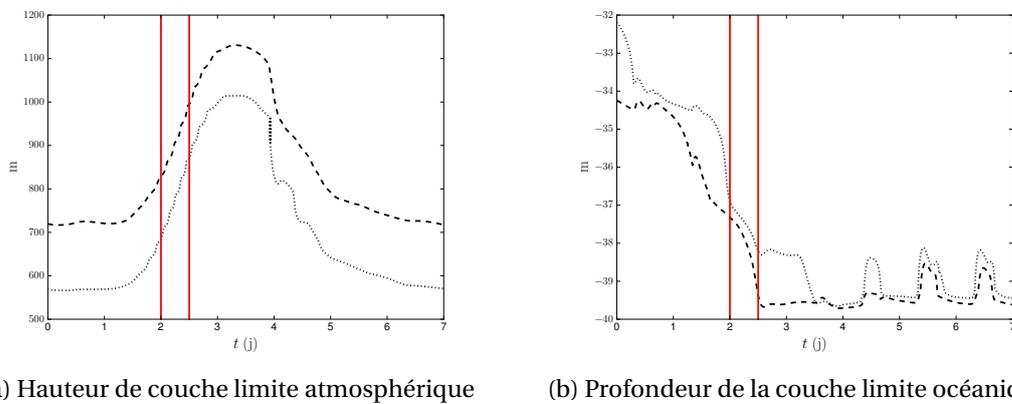


FIGURE 8.10 – (a) représente la hauteur de couche limite atmosphérique pour l'ébauche (pointillés) et l'état vrai (tirets), et (b) la profondeur de la couche limite océanique. La période temporelle délimitée par les lignes rouges correspond à notre future fenêtre d'assimilation.

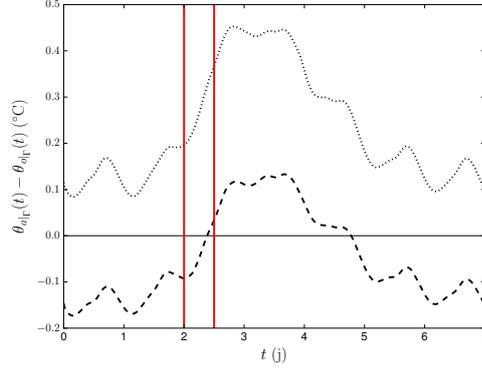


FIGURE 8.11 – Différence de température entre l’atmosphère et l’océan à l’interface air-mer. L’ébauche (en pointillés) est dans un régime stable tout au long de la prévision. La période temporelle délimitée par les lignes rouges correspond à notre future fenêtre d’assimilation.

Observations

Nous définissons les observations pour compléter le jeu de données qui sera utilisé dans nos expériences d’assimilation. Les observations sont définies à partir de l’état vrai précédemment introduit, telles que :

$$y_i^o = \mathbf{x}_i^t + \boldsymbol{\varepsilon}_i^o$$

où y_i^o est l’observation au temps t_i sur tout le domaine Ω , entachée d’une erreur $\boldsymbol{\varepsilon}_i^o$. Ces erreurs d’observations sont des bruits blancs gaussiens centrés en 0 :

$$\boldsymbol{\varepsilon}_i^o = \begin{pmatrix} \boldsymbol{\varepsilon}_{a,i}^o \\ \boldsymbol{\varepsilon}_{o,i}^o \end{pmatrix} \quad \text{avec} \quad \boldsymbol{\varepsilon}_{a,i}^o \sim \begin{pmatrix} \mathcal{N}(0, \sigma_{u_a}^o{}^2) \\ \mathcal{N}(0, \sigma_{v_a}^o{}^2) \\ \mathcal{N}(0, \sigma_{\theta_a}^o{}^2) \\ \mathcal{N}(0, \sigma_{q_a}^o{}^2) \end{pmatrix} \quad \text{et} \quad \boldsymbol{\varepsilon}_{o,i}^o \sim \begin{pmatrix} \mathcal{N}(0, \sigma_{u_o}^o{}^2) \\ \mathcal{N}(0, \sigma_{v_o}^o{}^2) \\ \mathcal{N}(0, \sigma_{\theta_o}^o{}^2) \\ \mathcal{N}(0, \sigma_{S_o}^o{}^2) \end{pmatrix}$$

où $\mathcal{N}(0, \sigma^2)$ désigne une loi normale centrée en 0 de variance σ^2 . Dans la suite, nous choisirons les variances suivantes pour définir le bruit des observations :

$$\begin{aligned} \sigma_{u_a}^o &= 3 \cdot 10^{-2} \text{ m.s}^{-1} & \sigma_{u_o}^o &= 7 \cdot 10^{-4} \text{ m.s}^{-1} \\ \sigma_{v_a}^o &= 3 \cdot 10^{-2} \text{ m.s}^{-1} & \sigma_{v_o}^o &= 7 \cdot 10^{-4} \text{ m.s}^{-1} \\ \sigma_{\theta_a}^o &= 2 \cdot 10^{-2} \text{ }^\circ\text{C} & \sigma_{\theta_o}^o &= 4 \cdot 10^{-2} \text{ }^\circ\text{C} \\ \sigma_{q_a}^o &= 2 \cdot 10^{-5} \text{ kg/kg} & \sigma_{S_o}^o &= 1.5 \cdot 10^{-2} \text{ psu} \end{aligned} \tag{8.1}$$

8.2 Description des expériences numériques

Maintenant que le jeu de données est complètement défini sur la fenêtre $[0, T]$, nous déterminons une sous-fenêtre temporelle de 12 h sur laquelle nous réaliserons plusieurs expériences jumelles d’assimilation de données. Cette période d’assimilation doit présenter des variations intéressantes des différents traceurs et de la dynamique du modèle. Nous entendons par « variations intéressantes » une période où l’état du modèle est changeant, avec si possible le passage d’un régime stable à instable ou inversement. Nous serons en effet plus à même de juger de la qualité d’un algorithme sur une telle période, plutôt que sur une où la dynamique du

modèle est très stable. C'est pourquoi nous choisissons de nous placer sur la fenêtre débutant au deuxième jour de prévision, soit la fenêtre [48 h, 60 h]. Nous commencerons par présenter les observations qui seront utilisées dans l'ensemble des expériences ainsi que les matrices de covariance d'erreur \mathbf{B} et \mathbf{R} . Puis nous décrirons ensuite les expériences envisagées avec la nomenclature associée.

Remarque 8.2.1

Notons qu'il peut exister un équilibre optimal différent entre les termes de la fonction coût pour chaque algorithme d'assimilation. En effet, tous ces algorithmes ne présentent pas la même fonction coût, les mêmes vecteurs de contrôles, le même modèle, etc. Nous ne connaissons cependant pas ces réglages optimaux a priori. Nous faisons donc le choix, afin de comparer les algorithmes, de faire varier uniquement le poids global de l'ébauche dans la fonction coût en fonction du nombre de variables contrôlées.

8.2.1 Choix des observations et covariances d'erreur

Observations

Nous choisissons de considérer des observations de la colonne complète océan-atmosphère (Ω) toutes les trois heures sur la fenêtre d'assimilation. Nous avons donc quatre « observations-colonnes » à 48 + 3 h, 6 h, 9 h et 12 h. À ces observations sont associées une même matrice de covariance d'erreur bloc diagonale \mathbf{R} , définie à partir des variances d'erreurs appliquées aux observations (8.1). Chaque bloc diagonal de \mathbf{R} correspond à une variable observée, et est donc défini tel que :

$$\mathbf{R}_c = \sigma_c^2 \cdot \mathbf{I}$$

où c désigne une variable observée ($u_a, v_a, \theta_a, q_a, u_o, v_o, \theta_o$ ou encore S_o). Puisque nous observons la colonne dans son intégralité, l'opérateur d'observations H est l'identité.

Covariance d'erreur de l'ébauche

Nous définissons la matrice \mathbf{B} bloc diagonale, où chaque bloc \mathbf{B}_c représente la matrice de covariance d'ébauche pour une variable contrôlée c . Ces matrices \mathbf{B}_c sont multi-diagonales, définies à partir d'une variance σ_c^b et d'une longueur de corrélation L_c . Notons que selon l'algorithme considéré, nous pourrions aussi bien contrôler l'état initial ($\in \Omega$) que l'interface océan-atmosphère ($\in \Gamma$). La longueur de corrélation pourra donc représenter une distance ou un temps de corrélation. Les coefficients $b_{c,i,j}$ des matrices \mathbf{B}_c sont définis tels que :

$$\begin{aligned} \text{contrôle d'un variable } c \in \Omega, \quad b_{c,i,j} &= \sigma_c^{b^2} \exp\left(-\frac{(z_i - z_j)^2}{2L_c^2}\right) \\ \text{contrôle d'un variable } c \in \Gamma, \quad b_{c,i,j} &= \sigma_c^{b^2} \exp\left(-\frac{(t_i - t_j)^2}{2L_c^2}\right) \end{aligned}$$

Dans la suite, nous considérerons les coefficients suivants pour le contrôle de l'état initial :

$$\begin{array}{llll}
\sigma_{u_{a,0}}^b = 4 \cdot 10^{-2} \text{ m.s}^{-1} & L_{u_{a,0}} = 25 \text{ m} & \sigma_{u_{o,0}}^b = 1 \cdot 10^{-3} \text{ m.s}^{-1} & L_{u_{o,0}} = 5 \text{ m} \\
\sigma_{v_{a,0}}^b = 5 \cdot 10^{-2} \text{ m.s}^{-1} & L_{v_{a,0}} = 25 \text{ m} & \sigma_{v_{o,0}}^b = 2 \cdot 10^{-3} \text{ m.s}^{-1} & L_{v_{o,0}} = 5 \text{ m} \\
\sigma_{\theta_{a,0}}^b = 5 \cdot 10^{-2} \text{ }^\circ\text{C} & L_{\theta_{a,0}} = 50 \text{ m} & \sigma_{\theta_{o,0}}^b = 5 \cdot 10^{-2} \text{ }^\circ\text{C} & L_{\theta_{o,0}} = 5 \text{ m} \\
\sigma_{q_{a,0}}^b = 3 \cdot 10^{-5} \text{ kg/kg} & L_{q_{a,0}} = 25 \text{ m} & \sigma_{S_{o,0}}^b = 2 \cdot 10^{-2} \text{ psu} & L_{S_{o,0}} = 5 \text{ m}
\end{array}$$

ainsi que les coefficients suivants lorsque l'on contrôlera l'interface :

$$\begin{array}{llll}
\sigma_{u_{a|\Gamma}}^b = 4 \cdot 10^{-2} \text{ m.s}^{-1} & L_{u_{a,0}} = 25 \text{ s} & \sigma_{u_{o|\Gamma}}^b = 1 \cdot 10^{-3} \text{ m.s}^{-1} & L_{u_{o,0}} = 0 \text{ s} \\
\sigma_{v_{a|\Gamma}}^b = 5 \cdot 10^{-2} \text{ m.s}^{-1} & L_{v_{a,0}} = 25 \text{ s} & \sigma_{v_{o|\Gamma}}^b = 1 \cdot 10^{-3} \text{ m.s}^{-1} & L_{v_{o,0}} = 0 \text{ s} \\
\sigma_{\theta_{a|\Gamma}}^b = 5 \cdot 10^{-2} \text{ }^\circ\text{C} & L_{\theta_{a,0}} = 50 \text{ s} & \sigma_{\theta_{o|\Gamma}}^b = 2 \cdot 10^{-1} \text{ }^\circ\text{C} & L_{\theta_{o,0}} = 10 \text{ s} \\
\sigma_{q_{a|\Gamma}}^b = 3 \cdot 10^{-5} \text{ kg/kg} & L_{q_{a,0}} = 25 \text{ s} & &
\end{array}$$

Remarque 8.2.2

La salinité de surface n'est pas nécessaire à l'évaluation des flux reçus par l'atmosphère. Nous ne contrôlerons donc pas ce champ. Notons également que le contrôle de l'interface côté atmosphère est nécessaire lorsque nous considérons deux modèles découplés.

Afin que le poids global de l'ébauche par rapport à celui des observations soit équivalent dans les différentes expériences, nous faisons le choix de multiplier J^b par un coefficient $\gamma_{\mathbf{B}}$. Ce coefficient différera selon le nombre de variables contrôlées, et sera d'autant plus faible que leur nombre sera élevé :

$$\gamma_{\mathbf{B}} = \frac{\#\mathbf{x}_0}{\#\mathbf{x}}$$

où $\#\mathbf{x}_0$ désigne la taille de l'état initial, et $\#\mathbf{x}$ désigne la taille du vecteur de contrôle. Si le vecteur de contrôle est composé uniquement de l'état initial, alors $\gamma_{\mathbf{B}} = 1$.

8.2.2 Contrainte faible de couplage

Il nous faut également expliciter le terme de contrainte faible J^s présent dans les fonctions coût de certains algorithmes. Cette contrainte faible minimise l'erreur de couplage. Si le modèle est fortement couplé, alors cette contrainte faible est nulle. En revanche, dans le cas d'un modèle couplé de manière asynchrone, nous pouvons écrire l'erreur de couplage ainsi :

$$(\text{couplage asynchrone}) \quad \|\mathcal{F}(\mathbf{x}_{a|\Gamma}^1, \mathbf{x}_{o|\Gamma}^0, \mathcal{R}) - \mathcal{F}(\mathbf{x}_{a|\Gamma}^1, \mathbf{x}_{o|\Gamma}^1, \mathcal{R})\| \quad (8.2)$$

où $\mathbf{x}_{o|\Gamma}^0$ désigne le first-guess à l'interface, et $\mathbf{x}_{a|\Gamma}^1$ et $\mathbf{x}_{o|\Gamma}^1$ sont les solutions à l'interface du modèle couplé de manière asynchrone ($k_{\max} = 1$). Dans le cas de deux modèles découplés, l'erreur de couplage s'écrira :

$$(\text{modèles découplés}) \quad \|\mathcal{F}(\mathbf{x}_{a|\Gamma}^{\text{unc}}, \mathbf{x}_{o|\Gamma}^0, \mathcal{R}) - \mathcal{F}(\mathbf{x}_{a|\Gamma}^0, \mathbf{x}_{o|\Gamma}^{\text{unc}}, \mathcal{R})\| \quad (8.3)$$

où $\mathbf{x}_{a|\Gamma}^0$ et $\mathbf{x}_{o|\Gamma}^0$ sont les first-guess à l'interface pour les deux modèles d'océan et d'atmosphère, et $\mathbf{x}_a^{\text{unc}}$ et $\mathbf{x}_o^{\text{unc}}$ sont les solutions des modèles découplés. Dans les deux cas ((8.2) et (8.3)), \mathcal{F} représente le calcul des différents flux de quantité de mouvement $\boldsymbol{\tau}$, de chaleur net Q_{net} et

d'eau douce F sur la fenêtre temporelle d'assimilation de données ([48, 60] h). Nous noterons $\mathcal{F}^{k,l} = \mathcal{F}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^l, \mathcal{R})$ tel que :

$$\mathcal{F}^{k,l} = \begin{pmatrix} \boldsymbol{\tau}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^l) \\ Q_{\text{net}}(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^l, \mathcal{R}) \\ F(\mathbf{x}_{a|\Gamma}^k, \mathbf{x}_{o|\Gamma}^l, \mathcal{R}) \end{pmatrix} = \begin{pmatrix} \boldsymbol{\tau}^{k,l} \\ Q_{\text{net}}^{k,l} \\ F^{k,l} \end{pmatrix}$$

La fonction coût J^s s'écrit alors, dans le cas d'un couplage asynchrone :

$$\text{(couplage asynchrone)} \quad J^s(\mathbf{x}_0) = \frac{1}{2} \gamma \left[\begin{pmatrix} \boldsymbol{\tau}^{1,0} \\ Q_{\text{net}}^{1,0} \\ F^{1,0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\tau}^{1,1} \\ Q_{\text{net}}^{1,1} \\ F^{1,1} \end{pmatrix} \right]^T \mathbf{W}^{-1} \left[\begin{pmatrix} \boldsymbol{\tau}^{1,0} \\ Q_{\text{net}}^{1,0} \\ F^{1,0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\tau}^{1,1} \\ Q_{\text{net}}^{1,1} \\ F^{1,1} \end{pmatrix} \right]$$

et dans le cas de deux modèles découplés :

$$\text{(modèles découplés)} \quad J^s(\mathbf{x}_0) = \frac{1}{2} \gamma \left[\begin{pmatrix} \boldsymbol{\tau}^{\text{unc},0} \\ Q_{\text{net}}^{\text{unc},0} \\ F^{\text{unc},0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\tau}^{0,\text{unc}} \\ Q_{\text{net}}^{0,\text{unc}} \\ F^{0,\text{unc}} \end{pmatrix} \right]^T \mathbf{W}^{-1} \left[\begin{pmatrix} \boldsymbol{\tau}^{\text{unc},0} \\ Q_{\text{net}}^{\text{unc},0} \\ F^{\text{unc},0} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\tau}^{0,\text{unc}} \\ Q_{\text{net}}^{0,\text{unc}} \\ F^{0,\text{unc}} \end{pmatrix} \right]$$

Covariances d'erreur de couplage

La matrice de covariance \mathbf{W} présente dans le terme J^s sera définie diagonale, telle que :

$$\mathbf{W} = \begin{pmatrix} \sigma_{\boldsymbol{\tau}}^2 \mathbf{I} & & \mathbf{0} \\ & \sigma_{Q_{\text{net}}}^2 \mathbf{I} & \\ \mathbf{0} & & \sigma_F^2 \mathbf{I} \end{pmatrix}$$

avec les covariances suivantes pour l'ensemble des expériences :

$$\sigma_{\boldsymbol{\tau}_{\text{zon}}} = 1 \cdot 10^{-3} \text{ m.s}^{-1} \quad \sigma_{Q_{\text{net}}} = 1 \text{ W.m}^{-2} \quad \sigma_F = 3 \cdot 10^{-6} \text{ m.s}^{-1}$$

Ces covariances d'erreur ont été choisies arbitrairement, en fonction des ordres de grandeur des différents flux d'interface. Enfin, nous considérerons également que $\tilde{\gamma} = \gamma$.

8.2.3 Algorithmes considérés

Nous souhaitons tester et comparer les différents algorithmes d'assimilation de données couplée introduits dans le chapitre 4. Tout d'abord, nous considérerons du côté des méthodes homogènes :

- MH-C : une expérience avec un modèle couplé à convergence de Schwarz (**MH-C-F**), une autre avec un modèle couplé de manière asynchrone (**MH-C-1**), en contrôlant l'état initial dans les deux cas.
- MH-CR : une expérience avec un modèle couplé de manière asynchrone en contrôlant l'état initial ainsi que le first-guess à l'interface (**MH-CR-Γ-1**).
- MH-CCF : une expérience avec deux modèles découplés en contrôlant l'état initial et l'interface entre les deux modèles (**MH-CCF-Γ**).

Puis nous considérerons les expériences suivantes du côté des méthodes inhomogènes :

- MI-C : trois expériences, une avec un modèle couplé à convergence de Schwarz (**MI-C-F**), une autre avec un modèle couplé de manière asynchrone (**MI-C-1**), et une dernière avec un modèle couplé de manière asynchrone, en couplant par itérations de Schwarz la première sous-fenêtre temporelle, grâce aux itérations de boucles externes (**MI-C-1-S**). Concrètement, nous corrigeons le first-guess de la première sous-fenêtre, à partir de la solution fournie par le modèle non-linéaire dans la boucle externe précédente.
- MI-CRCF : une expérience avec un modèle fortement couplé et le contrôle de l'état initial seulement (**MI-CRCF-F**), puis une expérience avec un modèle couplé de manière asynchrone et le contrôle de l'état initial ainsi que de l'interface (**MI-CRCF-1-Γ**).

Enfin, nous réaliserons une expérience d'assimilation découplée avec les deux modèles d'océan et d'atmosphère (**UNC**). Cette expérience nous servira simplement à évaluer le coût des méthodes couplées vis à vis d'une assimilation totalement découplée. Nous proposons de récapituler l'ensemble de ces expériences dans le tableau 8.1. Par chacune des expériences, nous pourrions envisager une dizaine d'itérations de boucle externe afin d'évaluer la convergence de chacun des algorithmes. Puis nous réaliserons une dernière comparaison pour un coût de calcul équivalent, avec deux itérations de boucle externe. L'ensemble de ces expériences devrait nous permettre d'évaluer les performances des méthodes inhomogènes vis à vis des méthodes homogènes, et d'observer l'impact de la présence de la contrainte faible de couplage.

Expérience	Algorithme	Modèle BE	Modèle BI	γ_B	γ
MH-C-F	MH-C	M^{cvg}	\mathbf{M}^{cvg}	1	-
MH-C-1	MH-C	$M^{k_{max}}, k_{max} = 1$	$\mathbf{M}^{k_{max}}$	1	-
MH-CR-Γ-1	MH-CR-Γ	$M^{k_{max}}, k_{max} = 1$	$\mathbf{M}^{k_{max}}$	1/1.45	1
MH-CCF-Γ	MH-CCF-Γ	M^{unc}	\mathbf{M}^{unc}	1/13.6	1
MI-C-F	MI-C	M^{cvg}	\mathbf{M}^{unc}	1	-
MI-C-1	MI-C	$M^{k_{max}}, k_{max} = 1$	\mathbf{M}^{unc}	1	-
MI-C-1-S	MI-C	$M^{k_{max}}, k_{max} = 1$	\mathbf{M}^{unc}	1	-
MI-CRCF-F	MI-CRCF	M^{cvg}	\mathbf{M}^{unc}	1.	0.01
MI-CRCF-1-Γ	MI-CRCF-Γ	$M^{k_{max}}, k_{max} = 1$	\mathbf{M}^{unc}	1/1.45	0.01
UNC	4D-Var incré.	M^{unc}	\mathbf{M}^{unc}	1	-

Tableau 8.1 – Récapitulatif des expériences réalisées avec le modèle simple colonne. Pour chaque expérience, nous décrivons l'algorithme utilisé, le modèle de la boucle externe (BE) et celui de la boucle interne (BI). Nous indiquons également la valeur des coefficients permettant de faire varier les poids de J^b et J^s au sein de la fonction coût.

Diagnostics

Avant de présenter les résultats, nous souhaitons définir quelques diagnostics qui seront réalisés. Comme dans le chapitre 5, nous commencerons par réaliser une comparaison globale de nos résultats à partir de plusieurs critères :

- le coût numérique,
- l'erreur de couplage à l'interface (ECI),
- amélioration par rapport à l'ébauche (ARE).

Le coût numérique sera évalué par rapport à l'assimilation découplée, en coût d'intégration du modèle non-linéaire et en coût d'intégration des modèles linéaire et adjoint. L'erreur de

couplage quantifie l'écart entre les flux perçus par l'atmosphère et l'océan. Si les modèles sont couplés en boucle externe alors nous avons :

$$ECI = \frac{1}{2} 10^{-5} \left(\mathcal{F}^{k,k-1} - \mathcal{F}^{k,k} \right)^T \mathbf{W}^{-1} \left(\mathcal{F}^{k,k-1} - \mathcal{F}^{k,k} \right)$$

où k représente la dernière itération du modèle couplé. En revanche, si les modèles sont découplés dans la boucle externe, alors cette erreur s'écrit :

$$ECI = \frac{1}{2} 10^{-5} \left(\mathcal{F}^{\text{unc},0} - \mathcal{F}^{0,\text{unc}} \right)^T \mathbf{W}^{-1} \left(\mathcal{F}^{\text{unc},0} - \mathcal{F}^{0,\text{unc}} \right)$$

où "unc" représente la solution après l'intégration du modèle découplé et 0 le first-guess à l'interface. Le facteur 10^{-5} est arbitraire. Enfin, nous évaluons l'amélioration par rapport à l'ébauche (ARE) comme un pourcentage d'amélioration vers l'état de référence. Considérons l'erreur quadratique rmse_c^a de l'analyse de la variable c par rapport à l'état vrai. Nous définissons alors ARE_c tel que :

$$\text{ARE}_c = \frac{\text{rmse}_c^b - \text{rmse}_c^a}{\text{rmse}_c^b} \leq 1$$

puis, nous calculons l'ARE en moyennant l'ensemble des ARE_c :

$$\text{ARE} = \frac{100}{8} \sum_{c_i} \text{ARE}_{c_i}$$

où les c_i représentent les huit variables du vecteur d'état des modèles d'océan et d'atmosphère. Le facteur 100 permet d'exprimer l'ARE comme un pourcentage d'amélioration global. Il est possible de distinguer trois cas :

- $\text{ARE} \in]0, 100]$, l'analyse a été améliorée par rapport à l'ébauche en moyenne sur $\Omega \times [0, T]$.
Notons que le cas $\text{ARE} = 100$ correspond à une analyse parfaite,
- $\text{ARE} = 0$, la qualité de l'analyse est équivalente à l'ébauche sur $\Omega \times [0, T]$,
- $\text{ARE} < 0$, l'analyse a été dégradée en moyenne par rapport à l'ébauche.

Remarque 8.2.3 - À propos du diagnostic ARE

Notons que ce diagnostic d'amélioration de l'analyse par rapport à l'ébauche est à considérer avec précaution. En effet, il s'agit d'un diagnostic global portant sur l'ensemble des variables sur l'intégralité du domaine d'intégration du modèle. Il ne permet pas d'attester des différentes erreurs commises par les méthodes, comme les chocs à l'initialisation. ARE nous permet donc d'avoir une première idée de la performance des algorithmes, mais il nous faudra observer les résultats plus en détails pour comparer les différentes méthodes.

8.3 Résultats et discussions

Nous présentons ici les résultats de l'ensemble des expériences citées dans le tableau 8.1. Nous effectuons tout d'abord dix itérations de boucle externe afin d'évaluer la performance des différents algorithmes à convergence, s'ils convergent. Puis nous nous restreignons ensuite à deux itérations de boucles externes afin d'évaluer la performance des algorithmes sous la contrainte d'un coût de calcul plus faible. Nous utilisons, comme dans le chapitre 5, la méthode du gradient conjugué préconditionné pour minimiser la fonction coût, avec comme critère de convergence $\varepsilon_{\text{PCG}} = 10^{-2}$ (5.3). Il s'agit d'un critère de convergence relativement faible, mais notons qu'en pratique, avec des modèles de prévision opérationnels, les itérations de minimisation sont stoppées avant d'atteindre ce critère.

8.3.1 Après 10 boucles externes

Le tableau 8.2 présente un premier diagnostic global des résultats des différentes expériences après dix itérations de boucle externe. Comme dans les expériences avec le modèle couplé linéaire (chapitre 5), la méthode MH-C-F présente de très bons résultats quant à la qualité du couplage et de la correction de l'ébauche. Mais nous observons également un coût de calcul élevé par rapport à l'assimilation découplée. Utiliser un modèle librement couplé permet d'améliorer ce coût numérique, mais dégrade alors la qualité du couplage à l'interface (expérience MH-C-1-10BE). Les différentes stratégies qui nous avaient semblé pertinentes dans le chapitre 5 sont alors évaluées.

- L'ajout d'une contrainte faible de couplage au sein d'une méthode homogène couplée (MH-CR- Γ -1-10BE) permet d'améliorer nettement le couplage à l'interface par rapport à une méthode sans contrainte faible (MH-C-1-10BE). En contrepartie, la méthode est moins performante pour retrouver l'état de référence, comme l'atteste l'ARE.
- Assurer le couplage entre les modèles par la seule contrainte faible (MH-CCF- Γ -10BE) permet également de produire une analyse qui soit mieux couplée à l'interface qu'avec la MH-C-1-10BE. En revanche, l'ébauche est encore moins bien corrigée qu'avec la MH-CR- Γ -1-10BE, puisque nous observons une ARE de 71% seulement.
- Les méthodes inhomogènes couplées sont très performantes dans cette configuration. En effet, pour un coût de calcul relativement faible, les MI-C-1-10BE et MI-C-1-S-10BE dépassent la méthode MH-C-F-10BE en terme d'ARE. En revanche, l'ECI dans ces expériences n'est pas aussi bon que pour les méthodes homogènes avec une contrainte faible de couplage.
- Enfin, les deux expériences avec une MI-CRCF présentent également de bonnes performances. Pour un coût de calcul équivalent, la MI-CRCF-1- Γ améliore l'ECI par rapport à la MI-C-1. La contrainte faible de couplage a donc un impact positif sur l'analyse produite. Ce n'est pas le cas où un modèle fortement couplé est utilisé, puisque la MI-CRCF-F présente des résultats légèrement moins bons que la MI-C-F.

Expérience	# itérations de minimisation	Coût modèle NL / UNC	Coût modèles TL-AD / UNC	ECI	ARE (%)
MH-C-F-10BE	152	3.6	5.3	$4.35 \cdot 10^{-13}$	86.5
MH-C-1-10BE	141	1	1.4	1.59	83.5
MH-CR- Γ -1-10BE	113	1	1.1	$2.78 \cdot 10^{-3}$	79
MH-CCF- Γ -10BE	183	1	1.8	$8.14 \cdot 10^{-3}$	71
MI-C-F-10BE	78	3.6	0.8	$7.98 \cdot 10^{-13}$	85
MI-C-1-10BE	128	1	1.3	$8.72 \cdot 10^{-1}$	87.5
MI-C-1-S-10BE	134	1	1.3	$7.91 \cdot 10^{-2}$	87
MI-CRCF-F-10BE	118	3.5	1.2	$1.03 \cdot 10^{-12}$	83.5
MI-CRCF-1- Γ -10BE	129	1	1.3	$2.08 \cdot 10^{-1}$	87.5
UNC-10BE	101	1	1	$1.76 \cdot 10^1$	59.5

Tableau 8.2 – Récapitulatif global des résultats des expériences d'assimilation couplée réalisées avec le modèle couplé non-linéaire, sur 10 itérations de boucles externes.

Ces premiers résultats montrent des performances équivalentes entre les différents algorithmes à convergence. Certains permettent une meilleure consistance des flux à l'interface tandis que

d'autres retrouvent mieux l'état de référence. De plus, mis à part les méthodes utilisant un modèle fortement couplé, toutes ces expériences ont un coût de calcul raisonnable par rapport à une assimilation découplée. Nous ne pouvons donc pas trancher sur le choix d'une de ces méthodes après dix itérations de boucles externes. Tout dépend si l'on souhaite privilégier le coût de calcul, la consistance de l'analyse à l'interface ou être le plus proche possible de l'état de référence. Nous pouvons ensuite nous intéresser à la convergence des méthodes inhomogènes. En effet, comme nous avons pu le mettre en évidence dans les chapitres 4 et 5 de ce manuscrit, les MI peuvent ne pas converger. L'ajout d'une contrainte faible peut alors permettre cette convergence.

Convergence des MI

La figure 8.12 présente la valeur de la fonction coût des MH-C et MI en fonction des itérations de boucle externe. Nous pouvons tout d'abord remarquer que toutes les méthodes convergent. Cependant, la fonction coût de la MI-C-F augmente lors de la deuxième itération de boucle externe. Ce choc est dû à l'approximation du modèle fortement couplé par deux modèles découplés. Cette approximation empêche l'algorithme de converger correctement, mais ce dernier arrive tout de même à « raccrocher » la convergence.

Remarque 8.3.1 - A propos de la convergence de la MI-C-F

Nous ne sommes pas toujours parvenu à « raccrocher » cette convergence au cours de nos expérimentations. En effet, la convergence de la MI-C-F est sensible à l'équilibre entre les différentes covariances d'ébauche et d'observation, et il est fréquent que cette méthode diverge.

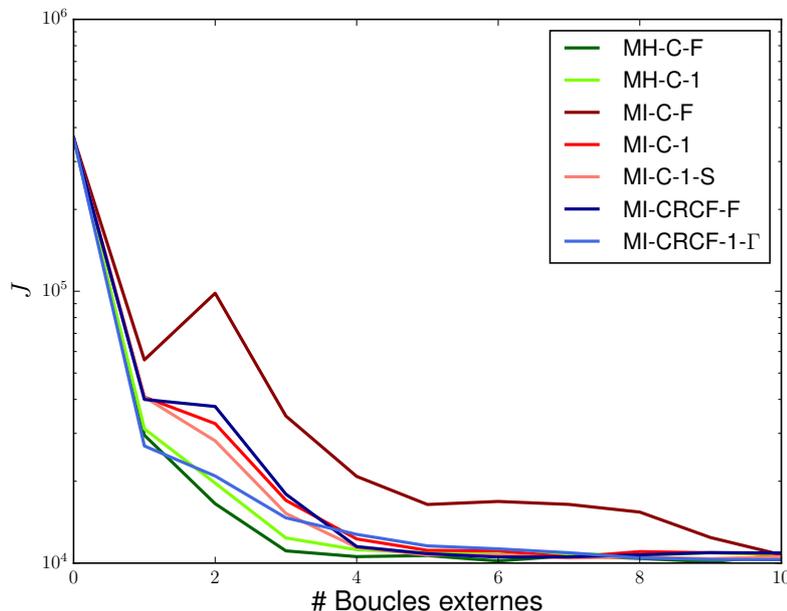


FIGURE 8.12 – Évolution de la valeur de la fonction coût en fonction des itérations de boucles externes pour les méthodes homogènes couplées, et les méthodes inhomogènes.

L'ajout d'une contrainte faible de couplage permet d'améliorer la convergence de la MI-C-F dans les premières itérations de boucles externes. Cela est montré par l'expérience MI-CRCF-

E, dont la fonction coût ne présente pas de pic à la deuxième itération de boucle externe. L'approximation d'un modèle couplé de manière asynchrone par deux modèles découplés ne semble en revanche pas poser de problème pour permettre la convergence des MI. Cela est montré par les expériences MI-C-1, MI-C-1-S et MI-CRCF-1- Γ . De plus, l'ajout d'une contrainte faible de couplage peut même améliorer la convergence dans les premières itérations de minimisation, comme nous pouvons le remarquer avec l'expérience MI-CRCF-1- Γ . En effet, la valeur de la fonction coût de cette dernière est plus faible que les autres MI sur les trois premières itérations de boucle externe.

Pour conclure, la convergence des MI est assurée pour deux raisons. Premièrement, nous partons d'une ébauche relativement proche de la solution du problème. Deuxièmement, le poids appliqué à l'ébauche (via la matrice \mathbf{B}) est proche de celui des observations. Ainsi, la convergence des MI est facilitée, la matrice \mathbf{B} permettant de ne pas trop s'écarter de la dynamique du système couplé. En effet, le critère de convergence des MI, explicité dans le chapitre 4, fait intervenir le rappel à l'ébauche ainsi que l'intégration des modèles découplés. De ce fait, si nous diminuons le poids de l'ébauche par rapport à celui des observations, il se peut que les MI divergent. Nous avons réalisé cette expérience avec la méthode MI-C-F, en multipliant les variances d'erreur d'ébauche d'un facteur 10, et en divisant par 10 les variances d'erreur des observations. La figure 8.13 montre que dans cette configuration, la MI-C-F diverge au bout d'une dizaine d'itérations de boucles externes. L'ajout d'une contrainte faible de couplage permet alors d'assurer la convergence de l'algorithme, comme nous avons pu le voir dans le cas d'un modèle linéaire dans le chapitre 5. Notons que le comportement de la MI-CRCF-F est tout de même erratique, et il n'est pas toujours assuré que cette méthode converge.

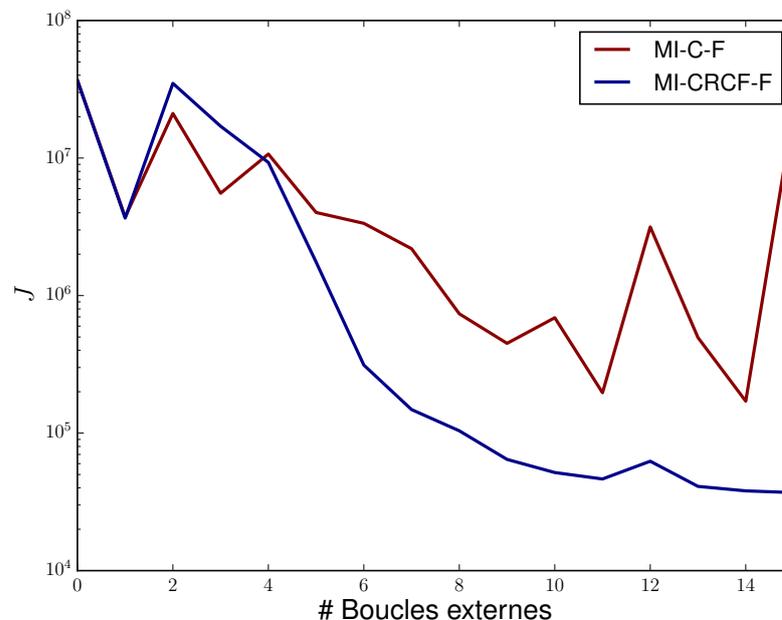


FIGURE 8.13 – Divergence de la méthode MI-C-F dans le cas où le poids de l'ébauche est trop faible par rapport à celui des observations. L'ajout de la contrainte faible de couplage permet de retrouver la convergence de l'algorithme.

Expérience d'une observation unique

L'ajout de la contrainte faible de couplage permet également la création de covariances d'erreur croisées entre les milieux à l'instant initial de la fenêtre d'assimilation. Pour mettre cela en évidence, une expérience classique en assimilation de données est de considérer une observation unique dans un des deux milieux. Nous observons donc l'océan au dernier pas de temps de la fenêtre d'assimilation au 6^{ème} niveau, soit à environ 8 m de profondeur. Notons que la matrice \mathbf{B} étant bloc diagonale, elle ne permet pas la création de covariances croisées à l'instant initial. La figure 8.14 présente les profils de l'incrément de température dans l'atmosphère, à l'instant initial de la fenêtre d'assimilation, pour les expériences présentant le terme J^s dans la fonction coût, ou utilisant des modèles linéaire et adjoint couplés lors de la minimisation.

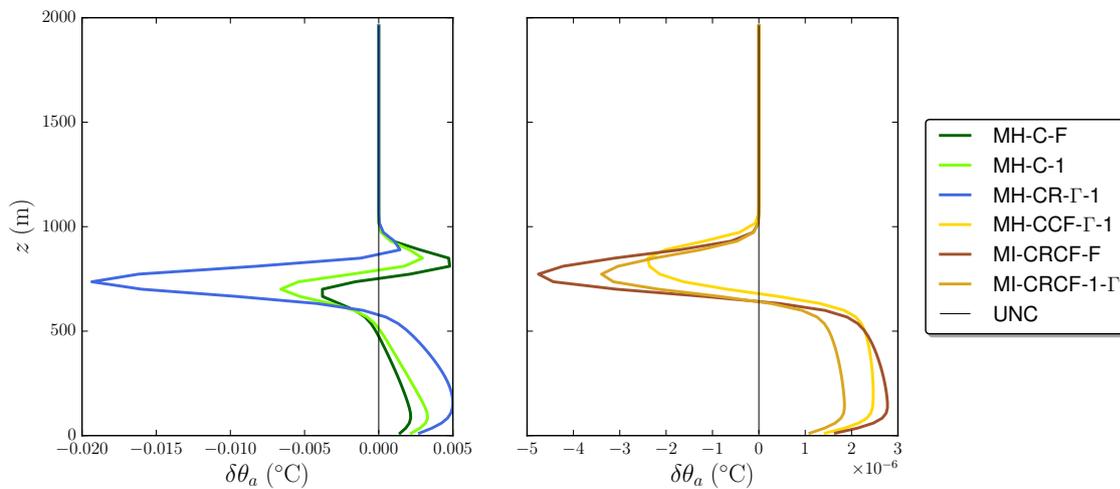


FIGURE 8.14 – Incrément d'analyse de la température atmosphérique au début de la fenêtre d'assimilation. A gauche, nous avons les méthodes homogènes couplés utilisant un modèle couplé lors de la minimisation. A droite, nous avons les méthodes homogènes et inhomogènes utilisant deux modèles découplés lors de la minimisation.

Le graphe de gauche dans la figure 8.14 concerne les méthodes homogènes avec un modèle couplé. Nous observons l'impact de l'intégration du modèle adjoint couplé sur l'incrément. En effet, le modèle adjoint couplé permet la création de covariances croisées entre les milieux, d'où un incrément initial non nul, à contrario de l'assimilation découplée. Nous remarquons que l'expérience MH-CR- Γ -1 présente un incrément plus fort que la MH-C-1. Cela s'explique par la présence de la contrainte faible J^s dans la fonction coût, qui permet de créer des covariances croisées plus fortes entre les deux milieux. L'impact de l'observation océanique est alors plus fort dans le milieu atmosphérique.

Le graphe de droite représente l'incrément créé par les méthodes n'utilisant pas de modèles linéaire et adjoint couplés lors de la minimisation. Le couplage entre les deux milieux lors de la minimisation est assuré par la seule contrainte faible J^s . Nous observons un incrément plus faible que dans les expériences avec le modèle couplé utilisé lors de la minimisation, de l'ordre de 10^{-6} . Il est cependant intéressant d'observer que la forme de l'incrément est la même, et est également identique à celle de l'incrément de l'état de référence par rapport à l'ébauche (non représenté ici). L'amplitude plus faible s'explique par le fait que le couplage est assuré par la seule contrainte faible.

Enfin, dans les deux graphes de la figure 8.14, l'incrément négatif visible au dessus de 750 m environ est significatif d'une modification de la hauteur de couche limite entre l'ébauche et l'analyse, dû à l'observation océanique. Les méthodes dont le couplage est assuré via la seule contrainte faible retrouvent la même hauteur de couche limite que les méthodes homogènes couplés, correspondant à la hauteur de couche limite de l'état de référence. Pour conclure, nous observons donc un impact positif de la contrainte faible de couplage lorsque nous n'avons pas de covariances croisées présentes dans la matrice \mathbf{B} . En effet, cette contrainte permet la création de covariances entre les milieux à l'état initial de la fenêtre d'assimilation, ce que ne permet pas les méthodes inhomogènes sans contrainte faible.

Conclusion partielle

A partir de ces résultats, nous ne pouvons pas discriminer une méthode par rapport à une autre dans la configuration actuelle de notre système. En effet, les différents algorithmes produisent des résultats équivalents à convergence. Certains sont moins coûteux mais présentent une analyse moins bien équilibrée à l'interface, d'autres se rapprochent plus de l'état de référence, etc. Nous avons cependant pu montrer que la convergence des méthodes inhomogènes n'était pas nécessairement assurée, et que l'ajout d'une contrainte faible de couplage pouvait aider à retrouver cette convergence. Cette contrainte peut également améliorer la performance d'une MI qui converge dans les premières itérations de boucles externes. De plus, nous avons pu observer que la contrainte faible de couplage permettait la création de covariances croisées entre les milieux lors de l'assimilation de données, notamment lorsque deux modèles découplés sont utilisés lors de la minimisation. Nous allons à présent estimer la performance de ces différents algorithmes avec une contrainte en terme de coût de calcul, c'est à dire en se limitant à deux itérations de boucle externe.

8.3.2 Après 2 boucles externes

Nous choisissons de restreindre les algorithmes à deux itérations de boucle externe, puisqu'il s'agit de la configuration utilisée par l'algorithme CERA au CEPMMT. Nous gardons pour ces expériences les mêmes observations, matrices de covariances et poids des termes dans la fonction coût, que pour celles présentées dans le tableau 8.2. Le tableau 8.3 récapitule les scores des différentes méthodes dans cette configuration. La méthode MH-C-F dépasse l'ensemble des méthodes en terme de qualité de l'analyse, mais est également la plus chère, comme cela pouvait être attendu. La conclusion principale qui ressort de ces résultats est la plus faible performance des méthodes inhomogènes sans contrainte faible par rapport aux autres. En effet, MI-C-1 et MI-C-1-S ont une ECI plus élevée et corrigent moins bien l'ébauche que les algorithmes MH-CR- Γ -1 et MH-CCF- Γ . L'expérience MI-C-F est même clairement en dessous puisque son ARE n'est que de 11%. Cela pouvait être attendu à la vue du comportement de la minimisation de la fonction coût, présenté dans la figure 8.12. Nous observons d'ailleurs que pour les MI, le rajout de la contrainte faible de couplage dans la fonction coût permet une nette amélioration après deux itérations de boucles externes. En effet, l'expérience MI-CRCF-F, équivalente à la MI-C-F avec le terme J^s en plus, a une ARE de 50.5%. De même, la MI-CRCF-1- Γ présente de meilleurs résultats que les MI-C-1 et MI-C-1-S. Cette première analyse des résultats met en avant l'impact positif que peut avoir la contrainte faible de couplage au sein des méthodes inhomogènes lorsque nous n'allons pas à convergence des boucles externes. Cette contrainte faible permet alors de se rapprocher de la qualité des résultats des méthodes homogènes.

Expérience	# itérations de minimisation	# exécutions du modèle NL	# exéc. des modèles TL-AD	ECI	ARE (%)
MH-C-F-2BE	22	3.5	4.1	$6.61 \cdot 10^{-13}$	81
MH-C-1-2BE	22	1	1	1.41	72.5
MH-CR- Γ -1-2BE	27	1	1.3	$3.07 \cdot 10^{-3}$	79
MH-CCF- Γ -2BE	29	1	1.4	$1.10 \cdot 10^{-2}$	74.5
MI-C-F-2BE	23	3.4	1.1	$1.26 \cdot 10^{-12}$	11
MI-C-1-2BE	24	1	1.1	$9.48 \cdot 10^{-1}$	64
MI-C-1-S-2BE	24	1	1.1	$7.45 \cdot 10^{-2}$	64.5
MI-CRCF-F-2BE	24	3.6	1.1	$7.94 \cdot 10^{-13}$	50.5
MI-CRCF-1- Γ -2BE	29	1	1.4	$2.48 \cdot 10^{-1}$	78
UNC-2BE	21	1	1	$1.80 \cdot 10^1$	43

Tableau 8.3 – Récapitulatif global des résultats des expériences d'assimilation couplée réalisées avec le modèle couplé non-linéaire, avec 2 itérations de boucles externes.

Nous proposons ensuite de réaliser d'autres diagnostics pour évaluer la qualité de l'analyse de chacune des méthodes. Nous pouvons tout d'abord nous intéresser à l'évaluation des hauteurs de couches limites. La figure 8.15 présente le cas de la couche limite océanique. Nous remarquons que l'ensemble des méthodes surestiment la profondeur de la couche, puis se réajustent ensuite. Les méthodes inhomogènes sans contrôle de l'interface (MI-C-F, MI-C-1, MI-C-1-S et MI-CRCF-F) parviennent moins bien que les autres à retrouver la bonne profondeur en fin d'intégration. Le contrôle de l'interface jointe à la contrainte faible de couplage dans l'expérience MI-CRCF-1- Γ permet d'améliorer l'évaluation de la profondeur de couche limite par rapport à MI-C-1 et MI-C-1-S, mais reste cependant moins bonne que pour les méthodes

homogènes. Les résultats quant à l'évaluation de la hauteur de couche limite atmosphérique sont similaires (figure non représentée).

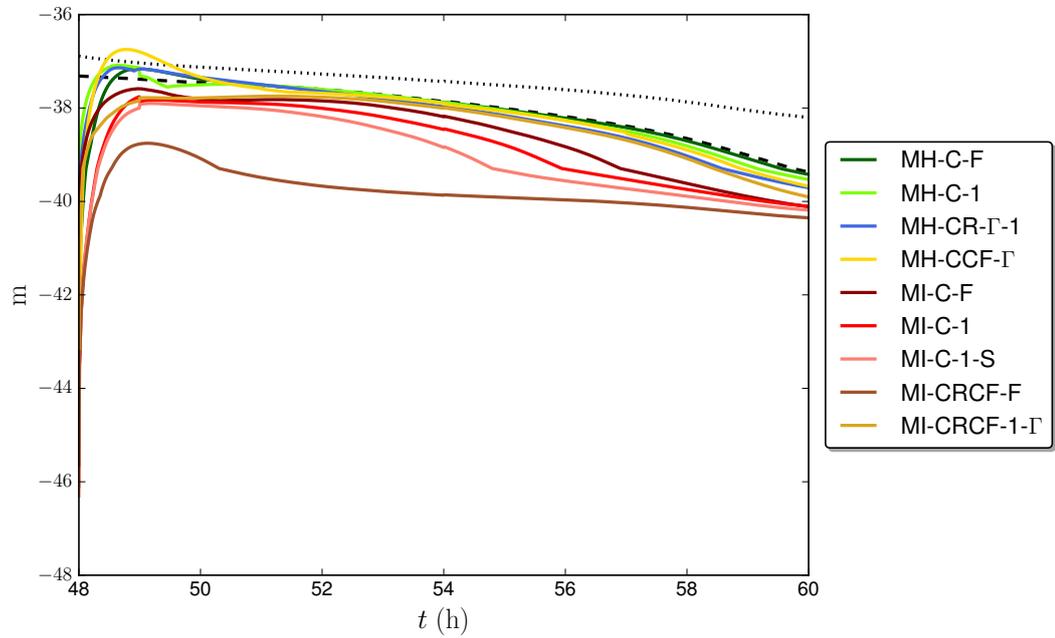


FIGURE 8.15 – Évolution de la profondeur de couche limite pour chacune des analyses produites. L'état vrai est représenté par des tirets et l'ébauche en pointillés.

Un second diagnostic intéressant est la capacité des algorithmes à retrouver le passage d'un régime stable à instable. En effet, l'ébauche se trouvait dans un régime stable sur toute la fenêtre d'assimilation alors que l'état de référence passe d'un régime instable à stable sur la fenêtre temporelle. La figure 8.16 représente l'évolution de la différence de température entre les deux milieux à l'interface. Nous observons alors que les méthodes ne parvenant pas à retrouver le régime instable en début de fenêtre temporelle, tendent ensuite à retrouver le bon régime grâce à la dynamique et les forçages du modèle. Nous ne pouvons donc pas conclure sur les performances des méthodes à retrouver les transitions de régimes

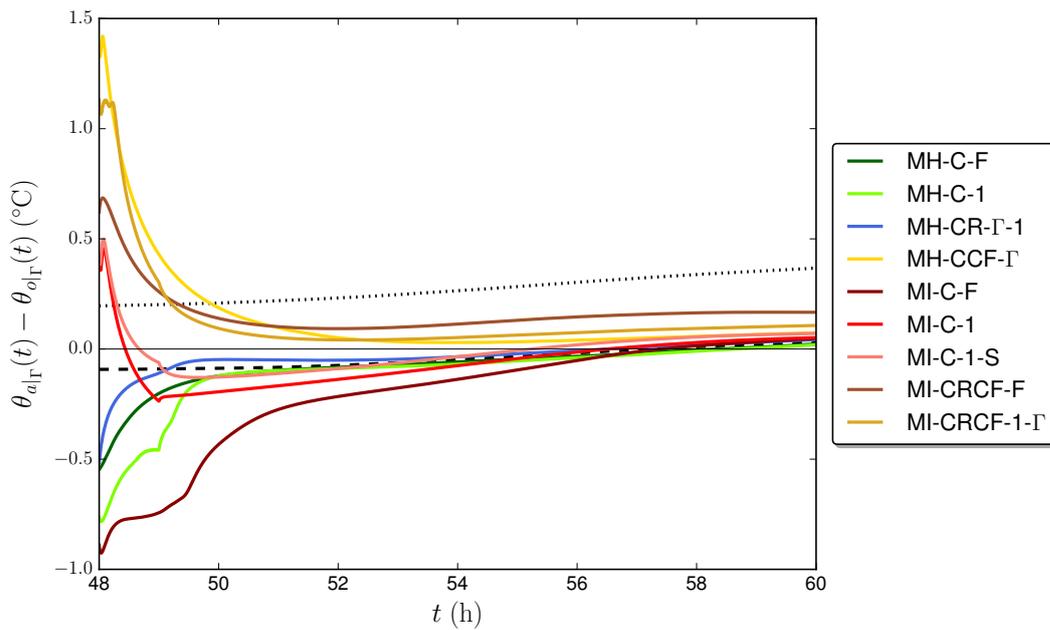


FIGURE 8.16 – Différence de température entre l'atmosphère et l'océan à l'interface air-mer, attestant d'un régime stable (si > 0) ou instable (si < 0) de l'atmosphère. L'état vrai est représenté par des tirets et l'ébauche en pointillés. Le trait plein noir permet de repérer le passage d'un régime à l'autre.

Enfin, nous nous intéressons à l'évolution des champs de surface, notamment la température atmosphérique, représentée par la figure 8.17. Nous pouvons observer sur cette figure deux chocs concernant l'évolution de $\theta_{a|\Gamma}(t)$. Le premier est à l'initialisation, et est présent dans les expériences MH-CCF- Γ , MI-C-1, MI-C-1-S, MI-CRCF-1- Γ , et dans une moindre mesure au sein de MI-C-F et MI-CRCF-F. Nous pouvons analyser ce choc comme étant dû à un déséquilibre de l'état initial par rapport à la dynamique du modèle. Ce dernier réajuste alors les différents champs lors des premiers pas de temps du modèle. Un second choc survient à $t = 49$ h, soit lors du premier changement de fenêtre temporelle. Il s'observe dans les expériences MH-C-1, MI-C-1, MI-C-1-S et MI-CRCF-1- Γ et est dû à l'utilisation d'un modèle couplé de manière asynchrone dans la boucle externe. En effet, le couplage asynchrone n'a pas permis aux différents champs de trouver leur équilibre au cours de la première fenêtre temporelle. Nous observons alors un réajustement du modèle suite au forçage de l'atmosphère par les flux évalués à partir de l'état de l'océan sur la fenêtre temporelle précédente. Ce second choc ne s'observe pas sur les autres expériences pour diverses raisons que nous pouvons citer.

- Dans le cas des expériences MH-C-F, MI-C-F et MI-CRCF-F, le modèle utilisé en boucle

externe est fortement couplé. L'état du modèle est alors en équilibre sur toute la fenêtre temporelle.

- Dans le cas de la MH-CR- Γ -1, le modèle est couplé de manière asynchrone mais le contrôle du first-guess ainsi que la contrainte faible de couplage permet de retrouver rapidement l'équilibre du modèle. De plus, la dynamique du modèle couplé au cours de la minimisation grâce à l'utilisation des modèles linéaire et adjoint couplés aide à trouver cet équilibre. En effet, l'adjoint du couplage permet de proposer un incrément d'analyse en équilibre avec les équations du modèle couplé. Nous n'observons donc pas de choc à l'interface dans les premiers pas de temps d'intégration du modèle au sein de cette expérience.
- Enfin, pour l'expérience MH-CCF- Γ , nous avons deux modèles découplés et nous contrôlons l'interface sur toute la fenêtre temporelle. Nous n'avons donc pas de choc à la fin de la première fenêtre temporelle puisque les modèles ne communiquent pas.

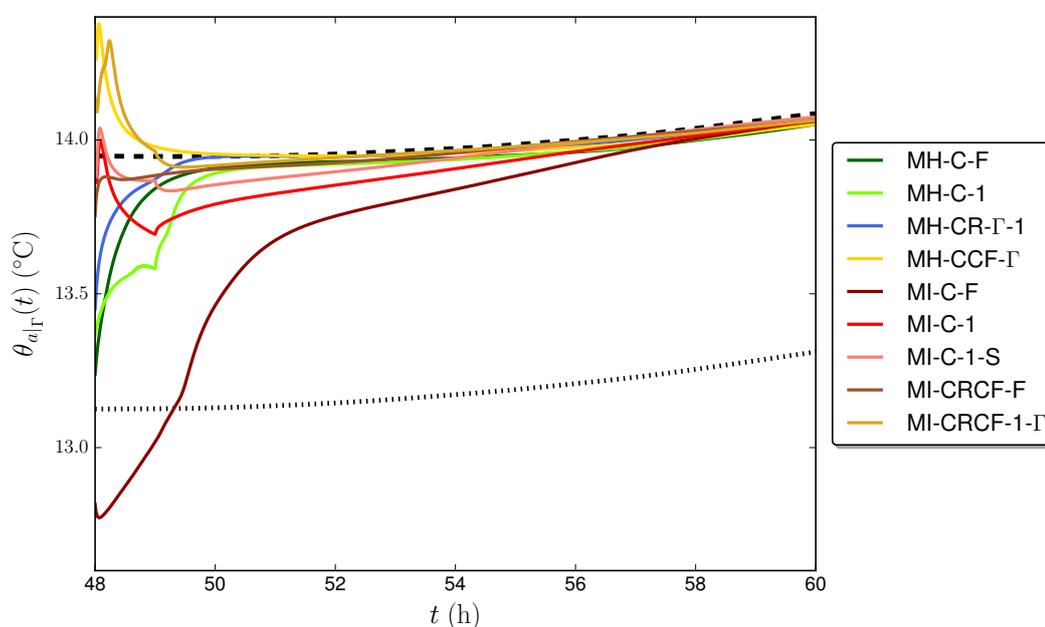


FIGURE 8.17 – Évolution de la température de surface pour chaque analyse. L'état vrai est représenté par des tirets et l'ébauche en pointillés.

Nous pouvons relier ces chocs à l'ECI, dont nous représentons l'évolution temporelle dans la figure 8.18 pour quelques méthodes. Le choc sur l'évolution de l'ECI à la fin de la première sous-fenêtre temporelle atteste d'un réajustement de l'ensemble du modèle couplé entre la première et deuxième fenêtre temporelle. Le modèle retrouve ensuite son équilibre à partir de la deuxième fenêtre. Nous remarquons que les chocs précédemment observés sur la température de surface sont corrélés aux chocs de l'évolution de l'ECI. Concernant la méthode MH-CR- Γ -1, nous observons également un réajustement de l'ECI à la fin de la première sous-fenêtre, mais ce dernier est très faible (10^{-5} , 10^{-6}) et n'impacte donc pas la température de surface. Enfin, un constat intéressant concerne les petits chocs de l'ECI présents pour la MH-CCF- Γ à 51, 54 et 57 h. Il s'agit des heures des observations, et ces chocs sont dus à la prise en compte de ces dernières par les deux modèles découplés. L'impact des observations déséquilibre alors légèrement les flux à l'interface. La contrainte faible J^s permet cependant de limiter ce déséquilibre.

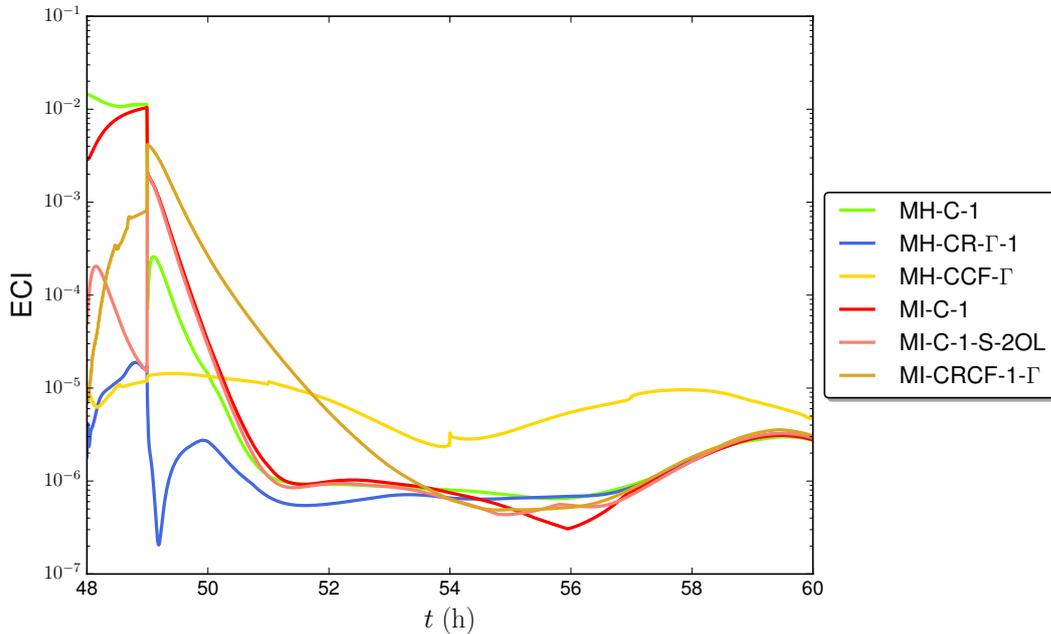


FIGURE 8.18 – Évolution de l'erreur de couplage à l'interface (ECI) au cours du temps, présenté en échelle logarithmique.

Conclusion partielle

Ces résultats permettent d'évaluer les différentes méthodes d'assimilation de données couplée lorsque nous considérons peu d'itérations de boucle externe. Limiter le nombre d'itérations est réalisé en pratique dans les centres de prévision opérationnels pour des raisons de coût. Nous avons pu alors observer que les méthodes inhomogènes ont plus de mal à converger rapidement, surtout si la différence entre le modèle couplé et ses deux composantes dé-couplées est importante. De même, l'utilisation d'un modèle couplé de manière asynchrone a certaines limites, et peut générer des chocs à l'interface comme nous avons pu le voir avec l'expérience MH-C-1. L'ajout d'une contrainte faible de couplage a alors un impact bénéfique pour limiter ces chocs. Nous avons également pu montrer le bon comportement des méthodes homogènes dont le couplage entre les modèles est assuré uniquement via le terme J^s au sein de la minimisation.

Prévisions

Pour terminer cette étude, nous proposons de réaliser une prévision à la suite de chaque analyse produite par les différentes expériences d'assimilation. Ces prévisions sont réalisées avec le même modèle, soit un modèle couplé de manière asynchrone. L'intégration est réalisée sur une fenêtre de 4 jours, commençant à $t = 60$ h, et découpée en 8 sous-fenêtres temporelles de 12 h. Le first-guess pour la première fenêtre temporelle est fourni par l'analyse utilisée pour la prévision, sur la fenêtre [48 h, 60 h]. Nous avons réalisé 9 prévisions dont nous allons analyser les résultats. Nous choisissons d'utiliser un modèle asynchrone, comme dans l'expérience MH-C-1, afin d'observer l'impact de la qualité du couplage de l'analyse lors de la prévision. Utiliser un modèle fortement couplé ne nous aurait pas permis cela. L'intérêt de réaliser une prévision post-assimilation est également de vérifier si le système prend en compte les cor-

rections de l'analyse ou si, au contraire, la qualité de l'analyse n'a pas d'impact majeur sur la qualité prévision.

Le tableau 8.4 présente deux premiers diagnostics. La prévision issue de l'analyse fournie par l'expérience MH-C-1 présente les meilleurs résultats, alors que cette analyse était parmi celles de moins bonne qualité en terme d'ECI et d'ARE. Nous expliquons cela par le fait que le même modèle est utilisé au sein de l'assimilation de données et de la prévision. En effet, Mulholland *et al.* (2015) on pu montrer que le changement de modèle entre l'assimilation et la prévision pouvait dégrader les prévisions. C'est pourquoi nous n'observons pas de meilleurs résultats à partir de l'analyse créée avec la MH-C-F.

Remarque 8.3.2 - Choix de la méthode d'assimilation

Cette première observation suppose qu'un choix de la méthode d'assimilation doit être fait en fonction de l'objectif recherché. Si nous souhaitons créer une ré-analyse, l'algorithme MH-C-F sera le meilleur candidat en terme de qualité de la ré-analyse. En revanche, si nous souhaitons créer une analyse pour ensuite réaliser une prévision, alors il sera préférable d'utiliser une méthode d'assimilation utilisant le modèle de la prévision.

Analyse pour la prévision	ECI	ARE (%)
MH-C-F-2BE	0.279	83
MH-C-1-2BE	0.295	92
MH-CR- Γ -1-2BE	0.287	83
MH-CCF- Γ -2BE	1.27	80
MI-C-F-2BE	0.560	63.5
MI-C-1-2BE	0.390	81
MI-C-1-S-2BE	0.383	81
MI-CRCF-F-2BE	0.728	74
MI-CRCF-1- Γ -2BE	0.898	79.5

Tableau 8.4 – Score des prévisions réalisées à partir de l'analyse produite par chaque expérience. Le modèle utilisé pour la prévision est un modèle couplé de manière asynchrone. La prévision est de 4 jours, et est décomposée en 8 fenêtres temporelles de 12 heures

La prévision MI-C-F présente les moins bons résultats en terme d'amélioration de l'ébauche. Cela était attendu au vu de la qualité de l'analyse que cette méthode a produite. Les autres prévisions issues des méthodes inhomogènes couplées présentent en revanche une ARE comparable aux méthodes homogènes. Nous pouvons ensuite observer une qualité de couplage plus faible pour les prévisions issues des expériences MH-CCF- Γ , MI-CRCF-F et MI-CRCF-1- Γ . Il s'agit des seules méthodes d'assimilation dont le couplage était assuré par la seule contrainte faible J^s lors de la minimisation. Cette erreur de couplage est en partie dû à un choc à l'initialisation que nous pouvons observer sur la figure 8.19. Ces trois expériences présentent le choc le plus fort dans les premiers pas de temps d'intégration du modèle couplé. Ce choc, provoqué par un déséquilibre entre l'état initial et l'interface, illustre les limites de la contrainte faible de couplage sur la qualité de l'analyse produite pour une prévision, lorsque cette dernière assure à elle seule le couplage. Enfin, nous remarquerons que les méthodes présentant un bon ECI (MH-C-F et MI-C-F) ne permettent pas de produire une prévision mieux couplée que les autres. Cela atteste que le système ne permet pas de tirer parti de la bonne qualité de couplage d'une analyse. La qualité de couplage de la prévision sera contrainte par la stratégie de couplage utilisée dans le modèle de la prévision.

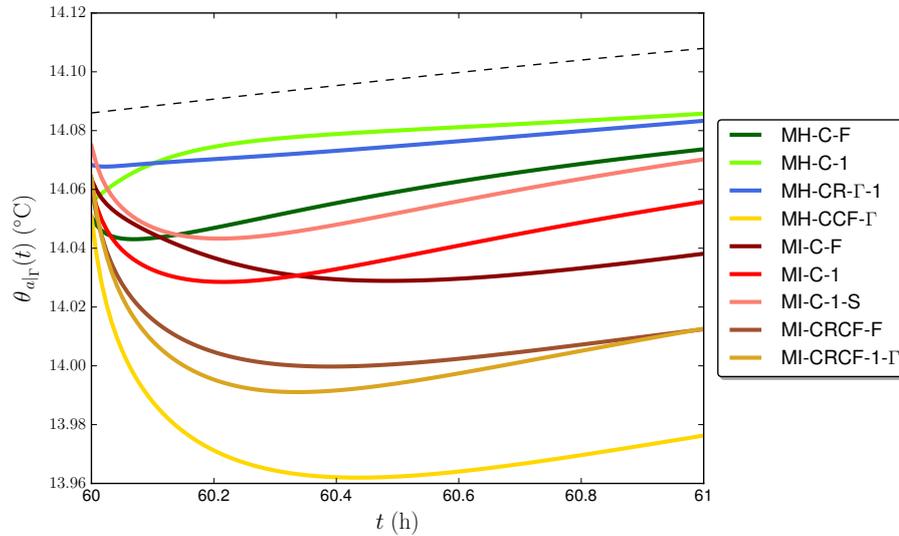


FIGURE 8.19 – Évolution de θ_{oI_r} durant la première heure d'intégration du modèle. Nous observons un choc à l'initialisation des modèles pour certaines prévisions, qui se réajustent ensuite pour revenir vers la trajectoire de référence (représenté par des tirets).

La contrainte faible de couplage a tout de même un impact global positif sur la qualité de la prévision. En effet, la figure 8.20 représente la prévision de la SSV. Nous remarquons que la prévision issue de la MI-C-F amplifie les variations de cette vitesse. Nous pouvons analyser cela comme un choc durant la prévision, dû à une mauvaise initialisation. La prévision issue de MI-CRCF-F présente en revanche une amplitude de variations beaucoup plus faible et se rapproche de la solution de l'état de référence. Nous observons ce phénomène dans une moindre mesure avec la prévision de la SSU (figure 8.21).

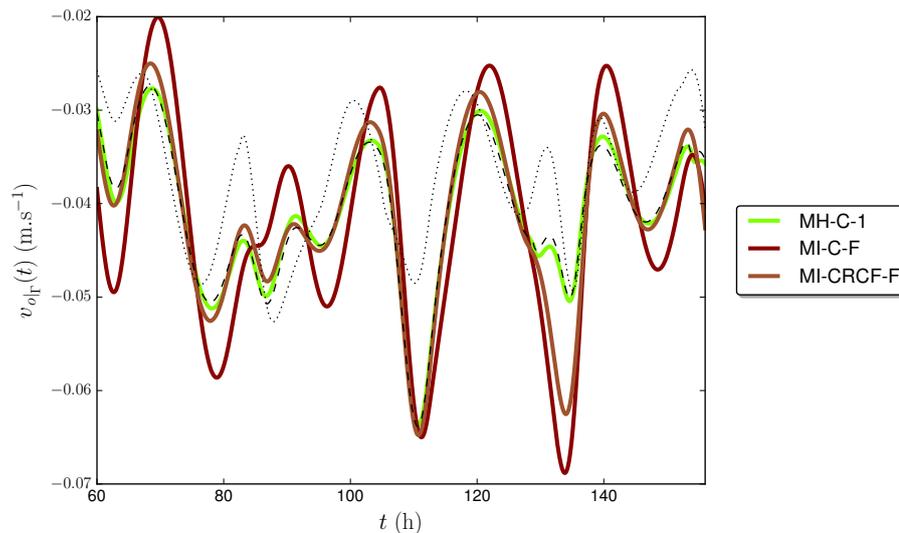


FIGURE 8.20 – Évolution de v_{oI_r} (SSV). La prévision issue de l'analyse fournie par la MI-C-F amplifie les variations de la SSV durant toute l'intégration du modèle. L'état vrai est représenté par des tirets et l'ébauche en pointillés.

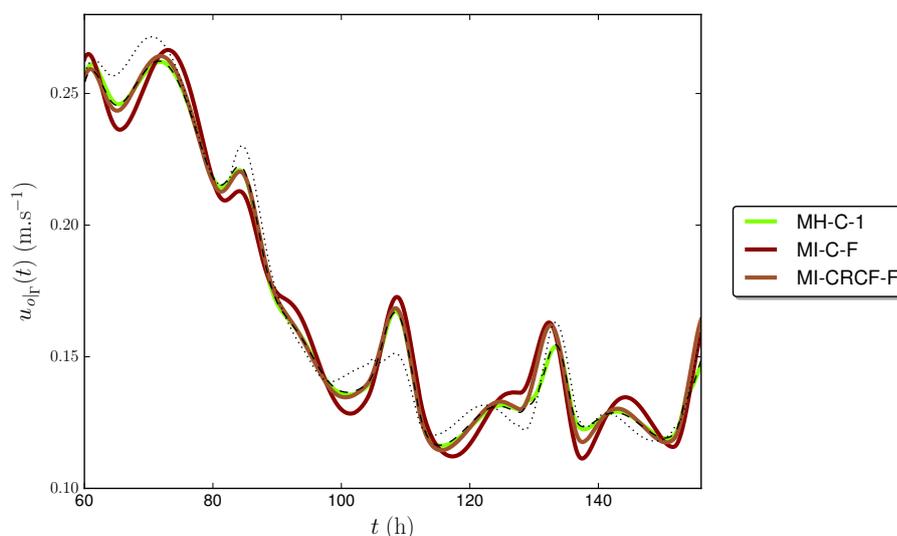


FIGURE 8.21 – Évolution de u_{olr} (SSU). La prévision issue de l'analyse fournie par la MI-C-F amplifie les variations de la SSU durant toute l'intégration du modèle. L'état vrai est représenté par des tirets et l'ébauche en pointillés.

Synthèse

Nous venons de réaliser plusieurs expériences à partir de quelques algorithmes d'assimilation de données couplée. Nous avons commencé par effectuer 10 itérations de boucle externe pour évaluer le comportement des méthodes à convergence. Nous avons pu alors observer qu'une méthode inhomogène couplée, utilisant un modèle fortement couplé en boucle externe, pouvait reproduire d'aussi bons résultats qu'une méthode homogène avec un modèle fortement couplé. Les méthodes inhomogènes avec un modèle couplée de manière asynchrone en boucle externe proposent quant à elles une bonne correction de l'ébauche, mais une qualité de couplage de l'analyse moins bonne. La convergence des méthodes inhomogènes n'est cependant pas forcément assurée, comme nous avons pu le mettre en évidence en diminuant le poids de l'ébauche par rapport à celui des observations. La contrainte faible de couplage peut alors permettre la convergence des MI, même si cette dernière n'est pas nécessairement gage de convergence. Nous avons également pu montrer l'impact positif que peut avoir cette contrainte pour la transmission d'informations d'un milieu à l'autre. Elle permet en effet la création de covariances croisées, permettant à une observation d'un milieu d'avoir un impact sur l'autre milieu. Enfin, la contrainte faible permettent d'améliorer la qualité du couplage, mais dégradent légèrement la correction de l'ébauche en contrepartie.

Nous avons ensuite évalué la performance des mêmes méthodes que précédemment, en s'arrêtant après deux itérations de boucles externes. Ce choix est justifié par le fait que les centres opérationnels ne réalisent en pratique que quelques itérations de boucles externes pour des raisons de coût, et nous pouvons par exemple citer le système CERA composé de deux itérations. Les méthodes inhomogènes sont plus impactées que les autres par cette contrainte de coût. En effet, ces dernières nécessitent un plus grand nombre d'itérations de boucle externe pour que la dynamique du modèle direct couplé puisse transmettre des informations d'un milieu à l'autre et ainsi créer une analyse de meilleure qualité. L'ajout d'une contrainte faible de couplage semble alors être bénéfique à ces méthodes. En effet, le terme J^s permet de prendre

en compte le couplage entre les modèles au cours de la minimisation. Enfin, nous avons pu démontrer la robustesse d'une méthode homogène dont le couplage entre les modèles est assuré uniquement via la contrainte faible lors de la minimisation. Une telle méthode à l'avantage non négligeable de ne pas avoir de modèle couplé à intégrer, permettant alors de gagner en temps de calcul. L'ajout du terme J^s au sein de la fonction coût peut cependant ralentir la minimisation.

Nous terminons ce chapitre par la réalisation d'une prévision avec un modèle couplé de manière asynchrone, à partir des analyses produites après deux itérations de boucle externe. Les résultats montrent que la qualité de couplage d'une analyse n'est pas un facteur déterminant pour la qualité de la prévision. En revanche, utiliser le même modèle pour la prévision et la création de l'analyse via l'assimilation de données a un impact fort sur la qualité de la prévision. Ainsi, la prévision réalisée à partir de l'analyse de la MH-C-1 est la plus proche de l'état de référence. Cette analyse était pourtant moins bien couplée, et était moins proche de l'état de référence que la plupart des autres analyses considérées. Nous remarquons donc, comme Mulholland *et al.* (2015), qu'il est préférable d'utiliser le même modèle au sein de l'assimilation de données et de la prévision. En revanche, il est plus judicieux d'utiliser une méthode comme la MH-CR- Γ -1 pour la création d'une réanalyse, cette dernière produisant une analyse de meilleure qualité pour un coût de calcul raisonnable.

Conclusions et perspectives

sur l'étude de quelques systèmes d'assimilation couplée

Conclusions

Ces travaux de thèse ont porté sur le développement et l'étude de quelques algorithmes d'assimilation de données adaptés aux modèles couplés océan-atmosphère. Il s'agit d'un domaine de recherche très actif ces dernières années, puisqu'il existe un réel besoin pour les centres de prévisions de disposer de tels algorithmes. En effet, de nombreux centres comme le CEPMMT développent actuellement leurs propres systèmes d'assimilation couplée. Cette thèse s'inscrit également dans le cadre du projet de recherche européen FP7 ERA-Clim2, au sein d'un *work package* portant sur le développement de nouvelles méthodes d'assimilation couplée pour la création de réanalyses du système terrestre.

Première partie - introduction au problème de l'assimilation de données couplée

Nous avons dans la première partie de ce manuscrit introduit les différents concepts nécessaires à la compréhension du problème de l'assimilation de données couplée. Nous avons commencé par présenter différentes méthodes d'assimilation, dont les méthodes variationnelles sur lesquelles sont basées ces travaux. Puis, nous avons introduit le problème du couplage dans les modèles d'océan et d'atmosphère. Quelques stratégies d'implémentations ont alors été explicitées, permettant d'obtenir des solutions dites *librement*, *faiblement* ou *fortement* couplées. Nous pouvons citer, entre autres, les méthodes asynchrones qui sont très utilisées dans les centres opérationnels. Ces méthodes produisent une solution librement couplée. Enfin, nous avons introduit les méthodes itératives de Schwarz, permettant d'obtenir une solution couplée au sens fort.

La compréhension de ces deux domaines de recherche, couplage et assimilation, nous a enfin permis d'introduire différentes méthodologies existantes dans la littérature pour assimiler des données dans un système couplé. Nous avons pu évaluer la qualité du couplage au sein des différents systèmes considérés, et sa prise en compte dans le processus d'assimilation. Il en ressort qu'aucune de ces études ne considère un modèle fortement couplé et ne propose une analyse couplée au sens fort. Cela est justifié par le fait que, pour des raisons de coût de calcul, la majorité des systèmes de prévisions couplés existants le sont librement, par des méthodes asynchrones. Nous avons alors relevé que le vocabulaire de l'assimilation de données couplée pouvait être ambigu. En effet, bien que l'ensemble des méthodes produise des analyses libre-

ment couplée, les systèmes d'assimilation peuvent être qualifiés de « fortement couplés ». Cela fait référence à la prise en compte du couplage du modèle au sein du processus d'assimilation de données, et non pas à la qualité du couplage.

A la suite de cet état de l'art, nous avons souhaité introduire quelques alternatives aux méthodes d'assimilation couplée existantes. Notre but est d'une part, d'introduire un nouveau vocabulaire pour classifier les systèmes d'assimilation couplée, moins ambigu concernant la qualité du couplage. Puis, d'autre part, proposer des méthodes d'assimilation permettant de produire des analyses plus fortement couplées que les méthodes existantes, pour un coût de calcul raisonnable.

Seconde partie - étude de quelques algorithmes et première application numérique

Nous proposons dans la seconde partie de ce manuscrit une nouvelle classification de méthodes variationnelles couplées en introduisant le concept d'assimilation de données couplée homogène et inhomogène. « Homogène » fait référence au fait que l'on utilise la linéarisation et l'adjoint du modèle non-linéaire utilisé dans la boucle externe du 4D-Var incrémental. « Inhomogène » désigne les algorithmes où un modèle couplé non linéaire est considéré dans la boucle externe, et deux modèles découplés linéaire et adjoint sont utilisés au sein de la boucle interne du 4D-Var incrémental. Nous proposons alors plusieurs algorithmes d'assimilation basés sur ces deux types de méthodes, visant à répondre au problème de l'assimilation couplée. Nous avons pour cela introduit un terme de pénalisation des erreurs de couplage entre les modèles au sein de la fonction coût. Ce terme de pénalisation vise à réduire les inconsistances de flux à l'interface lorsqu'un modèle faiblement, librement ou non couplé est utilisé.

Nous avons ensuite étudié la convergence de ces différents algorithmes. La convergence des algorithmes homogènes a déjà pu être étudiée dans la littérature, puisqu'il s'agit ni plus ni moins de méthodes variationnelles classiques. En revanche, la convergence des méthodes inhomogènes n'est pas assurée de part l'inconsistance des modèles entre les boucles interne et externe. Nous avons pu rapprocher ces méthodes inhomogènes aux méthodes dites de Gauss-Newton perturbées, dont la convergence a été étudiée par Gratton *et al.* (2007). Il ressort de leur étude un critère suffisant permettant la convergence de ces méthodes. Ce critère quantifie la différence entre la linéarisation exacte du modèle utilisé en boucle externe et la linéarisation approchée utilisée en boucle interne. Si cette différence est trop grande, alors l'algorithme risque de ne pas converger. Enfin, nous avons obtenu un critère nécessaire et suffisant de convergence des méthodes inhomogènes dans le cas où le modèle utilisé en boucle externe est linéaire. Ce critère quantifie également la différence entre la linéarisation exacte et approchée. De plus, nous avons pu quantifier l'erreur de la solution produite par ces méthodes inhomogènes, par rapport à la solution de référence (la solution du problème de la boucle externe).

Une première étude numérique des algorithmes d'assimilation couplée précédemment introduits a été menée dans un cadre linéaire. Nous avons pu évaluer la performance de ces différentes méthodes dans un cadre d'application relativement simple, avec un modèle de diffusion linéaire. Nous comparons les résultats à ceux d'une méthode homogène utilisée avec un modèle fortement couplée. Cette méthode produit l'analyse de meilleure qualité sur le plan du couplage et de l'erreur moyenne des champs par rapport à l'état de référence. Au sein de cette comparaison, les méthodes inhomogènes avec deux itérations de boucles externes présentent de moins bons résultats, en terme de restitution de l'état vrai et de qualité de couplage, que leurs homologues homogènes. Elles ont cependant l'avantage d'être moins coûteuses numériquement. Nous noterons en revanche qu'en cas de convergence, avec dix itérations de boucles externes, une méthode inhomogène avec une correction de l'interface à chaque itération de

boucle externe, à la manière des algorithmes de couplage de Schwarz, permet de produire une analyse avec une bonne qualité de couplage, et une erreur moyenne de l'analyse par rapport à l'état vrai très faible. Nous avons également pu mettre en avant l'impact positif que peut avoir la contrainte faible de couplage au cours de l'assimilation de données sur la convergence des méthodes inhomogènes. En effet, si le « choc » entre les boucles externes et interne est trop fort, alors cette contrainte peut permettre de raccrocher la convergence. En cas de convergence, la contrainte faible présente aussi un intérêt, celui de diminuer l'erreur commise par rapport à la solution de référence.

Troisième partie - applications à un système complexe

Ces seuls résultats ne permettent cependant pas de tirer de conclusions sur la pertinence des différents algorithmes introduits pour des systèmes de prévision opérationnels. En effet, un modèle réaliste présente des non-linéarités et non-différentiabilités qui rendent le processus d'assimilation de données plus compliqué. C'est pourquoi nous avons développé un modèle couplé 1D vertical non-linéaire, avec des paramétrisations sous-maille proche de l'interface. Ce modèle, dit « simple colonne », est alors plus représentatif du comportement des modèles de prévision opérationnels, sans en supporter le coût numérique. Ce type de modèle est classiquement utilisé pour étudier le comportement à l'interface des modèles réalistes. Nous avons implémenté ce modèle couplé au sein d'un surcouche logicielle, OOPS, développée au CEPMMT. Il s'agit d'un logiciel permettant la gestion de systèmes de prévision et d'assimilation de données. Cela nous permet, en plus de profiter des performances de ce logiciel, de pouvoir partager notre code au sein du projet ERA-Clim2, et de manière plus large à la communauté d'assimilation de donnée couplée et d'étude sur les systèmes couplés.

Plusieurs algorithmes d'assimilation couplée ont ainsi pu être testés avec ce modèle simple colonne. Il ressort de cette étude que les méthodes inhomogènes sont moins performantes que leurs homologues homogènes sous la contrainte d'une limitation de coût de calcul (deux itérations de boucle externe). Nous observons une convergence plus difficile que précédemment avec le modèle de diffusion linéaire. Nous avons alors pu mettre en avant l'intérêt majeur que représente dans ce cas la contrainte faible de couplage. En effet, cette contrainte permet une meilleure convergence des algorithmes, réduisant le « choc » entre les deux boucles. De plus elle permet la création de covariances croisées entre les milieux lorsque deux modèles découplés sont utilisés lors de la minimisation. Dans certaines configurations, cette contrainte permet également d'assurer la convergence des méthodes inhomogènes qui divergent autrement. Enfin, la robustesse d'une méthode homogène, où le couplage est assurée dans la minimisation par la seule contrainte faible, a été montrée. Une telle méthode présente l'avantage non négligeable de ne pas avoir le modèle couplé à intégrer, permettant ainsi un gain de coût de calcul. Cependant, la réalisation d'une prévision couplée à la suite d'une analyse produite par cette méthode peut présenter quelques limitations, dues au choc de changement de modèle entre l'assimilation et la prévision (Mulholland *et al.*, 2015).

L'ensemble de ces résultats permettent de progresser dans la conception d'algorithmes d'assimilation de données couplée performants, et sur la compréhension du problème dans son ensemble. Certains points nécessitent cependant d'être encore abordés en perspective de cette thèse.

Perspectives

Bien que nous ayons étudié théoriquement et numériquement le comportement de quelques algorithmes d'assimilation de données couplée, il reste encore beaucoup de travail dans ce domaine de recherche afin de proposer des algorithmes performants numériquement, permettant de produire des analyses avec une bonne consistance des flux à l'interface, et permettant la transmission d'informations d'un milieu à l'autre.

Concernant les algorithmes que nous avons pu présenter, il serait intéressant de réaliser d'autres diagnostics et expériences afin de mieux comprendre le comportement de chacun d'entre eux, et l'impact de la contrainte faible de couplage. Pour cela, il pourrait être judicieux d'utiliser un système plus complexe, tel que le modèle couplé utilisé par Laloyaux *et al.* (2015) pour produire la réanalyse couplée CERA-20C. Dans nos expériences, l'impact de la contrainte faible et du contrôle de l'interface étaient limités dans le cas des méthodes inhomogènes. En effet, l'information fournie par la minimisation concernant l'interface n'était utile que pour la première fenêtre temporelle. Ensuite, pour les autres fenêtres, la qualité du couplage à l'interface était principalement contrainte par la stratégie de couplage considéré. Il pourrait donc être intéressant de contrôler une correction des champs à l'interface sur toute la fenêtre, plutôt que l'interface elle-même sur la première fenêtre temporelle. Une telle stratégie pourrait permettre d'améliorer la consistance de l'analyse à l'interface.

Un autre point qui nous paraît important est le développement de la matrice de covariance d'erreur d'ébauche \mathbf{B} . En effet, nous avons considéré dans nos expériences des matrices \mathbf{B} bloc diagonales seulement, sans covariances croisées entre les milieux d'océan et d'atmosphère. Or, il nous semble important de pouvoir représenter le couplage entre les modèles au sein de cette matrice, afin de renforcer la consistance à l'interface de l'analyse produite. Nous pourrions évaluer ces covariances croisées à partir des conditions d'interface entre les deux modèles, afin d'assurer un équilibre de la même manière que nous le réalisons avec la contrainte faible J^s .

Enfin, nous avons dans ce manuscrit proposé un critère de convergence nécessaire et suffisant des méthodes inhomogènes dans un cadre linéaire. Il serait intéressant d'étendre cette étude au cadre non linéaire. Gratton *et al.* (2007) ont proposé une condition suffisante pour la convergence de ces méthodes dans un cadre non-linéaire, mais nous avons pu remarquer lors de nos expériences que cette condition était trop restrictive. En effet, de nombreuses expériences ne respectaient pas ce critère tout en convergeant. Il serait donc intéressant de trouver une condition, si ce n'est nécessaire et suffisante, plus pertinente que celle de Gratton *et al.* (2007). Une piste envisagée pour évaluer ce critère de convergence est de repartir de la démonstration de convergence du 4D-Var incrémental, pour ensuite évaluer l'impact de l'approximation du modèle linéaire couplé en deux modèles linéaires découplés au sein de la boucle interne. Il n'est cependant pas dit que nous arrivions déterminer un critère plus pertinent que celui de Gratton *et al.* (2007). À défaut, nous avons pensé à développer un 4D-Var incrémental à deux niveaux de boucles internes pour résoudre les problèmes inhomogènes. Le principe de cet algorithme est le suivant :

- la boucle externe est composée de la fonction coût non linéaire que l'on cherche à minimiser, avec le modèle couplé non-linéaire,
- la première boucle interne est la stricte linéarisation de la boucle externe, avec la linéarisation du modèle couplé,
- la seconde boucle interne, que l'on peut appeler « super-interne », est composée des deux composantes linéaires découplées du modèle couplé linéaire.

Nous proposons de schématiser cet algorithme dans la figure III. Dans ce schéma, J_{int} est la linéarisation de la fonction coût J en boucle externe, et \tilde{J}_{int} est son approximation. Le but est alors d'estimer l'incrément optimal $\delta \mathbf{x}_0^{(p)}$ de chaque boucle externe par itérations successives, avec une seconde boucle interne présentant un problème simplifié de la boucle interne. Dans cet algorithme, nous notons p les itérations de la première boucle interne, et q les itérations de la seconde. Nous définissons alors $\mathbf{x}_0^{(p,0)} = \mathbf{x}_0^{(p)}$ et que :

$$\begin{aligned} J_{\text{int}}(\delta \mathbf{x}_0^{(p)}) &= \frac{1}{2} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right)^T \mathbf{B}^{-1} \left(\delta \mathbf{x}_0^{(p)} + \sum_{l=0}^{p-1} \delta \mathbf{x}_0^{(l)} \right) + \frac{1}{2} \left(\mathbf{G} \delta \mathbf{x}_0^{(p)} - d^{(p)} \right)^T \mathbf{R}^{-1} \left(\mathbf{G} \delta \mathbf{x}_0^{(p)} - d^{(p)} \right) \\ &= \frac{1}{2} \left(\mathbf{x}_0^{(p)} - \mathbf{x}_0^b \right)^T \mathbf{B}^{-1} \left(\mathbf{x}_0^{(p)} - \mathbf{x}_0^b \right) + \frac{1}{2} \left(\mathbf{G} \mathbf{x}_0^{(p)} - (\mathbf{G} \mathbf{x}_0^b + d^{(p)}) \right)^T \mathbf{R}^{-1} \left(\mathbf{G} \mathbf{x}_0^{(p)} - (\mathbf{G} \mathbf{x}_0^b + d^{(p)}) \right) \end{aligned}$$

puis, en considérant les itérations de la boucle super-interne, nous avons que :

$$\begin{aligned} J_{\text{int}}(\mathbf{x}_0^{(p,q)}) &= \frac{1}{2} \left(\mathbf{x}_0^{(p,q)} - \mathbf{x}_0^b \right)^T \mathbf{B}^{-1} \left(\mathbf{x}_0^{(p,q)} - \mathbf{x}_0^b \right) + \frac{1}{2} \left(\mathbf{G} \mathbf{x}_0^{(p,q)} - (\mathbf{G} \mathbf{x}_0^b + d^{(p)}) \right)^T \mathbf{R}^{-1} \left(\mathbf{G} \mathbf{x}_0^{(p,q)} - (\mathbf{G} \mathbf{x}_0^b + d^{(p)}) \right) \\ &= \frac{1}{2} \left(\mathbf{x}_0^{(p,q)} - \mathbf{x}_0^b \right)^T \mathbf{B}^{-1} \left(\mathbf{x}_0^{(p,q)} - \mathbf{x}_0^b \right) + \frac{1}{2} \left(\mathbf{G} \mathbf{x}_0^{(p,q)} - \tilde{\mathbf{y}}^{o,(p)} \right)^T \mathbf{R}^{-1} \left(\mathbf{G} \mathbf{x}_0^{(p,q)} - \tilde{\mathbf{y}}^{o,(p)} \right) \end{aligned}$$

et $\mathbf{x}_0^{(p,q+1)} = \mathbf{x}_0^{(p,q)} + \delta \tilde{\mathbf{x}}_0^{(q)}$ avec $\delta \tilde{\mathbf{x}}_0^{(q)}$ l'incrément estimé par la boucle super-interne. Grâce à cet algorithme, il est possible de contrôler la condition de convergence la boucle super-interne vers la solution de la boucle interne à partir du critère de convergence que nous avons mis en évidence dans ce manuscrit. La convergence de l'algorithme dans son ensemble repose ensuite sur la convergence des itérations de boucle interne vers la solution de la boucle externe. Il s'agit de la convergence classique d'un 4D-Var incrémental, modulo l'erreur commise pour évaluer $\delta \mathbf{x}_0^{(p)}$ avec la boucle super-interne, que nous pouvons quantifier. L'inconvénient majeur de cette méthode est l'ajout d'une nouvelle boucle, ce qui augmente le coût de calcul par rapport à un 4D-Var incrémental classique.

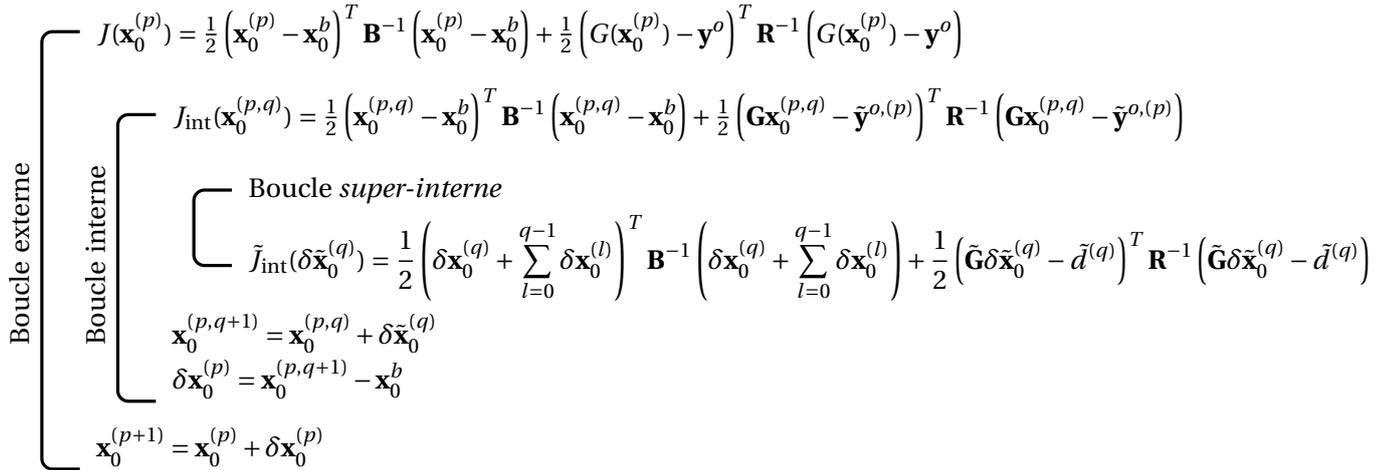


FIGURE III – Idée d'un algorithme d'assimilation couplé inhomogène à deux niveaux de boucle interne. Dans cet algorithme, on définit $\mathbf{x}_0^{(p,0)} = \mathbf{x}_0^{(p)}$ et $\tilde{\mathbf{y}}^{o,(p)} = \mathbf{G} \mathbf{x}_0^b + d^{(p)}$. $\tilde{\mathbf{G}}$ représente l'opérateur généralisé d'observation avec les deux modèles découplés d'océan et d'atmosphère. J_{int} désigne la fonction coût de la boucle interne, avec le modèle couplé linéaire, et \tilde{J}_{int} son approximation avec les deux modèles découplés.

Annexes

Annexe A

Liste des classes d'interfaces de OOPS et de leurs méthodes à implémenter

Le tableau A.1 liste l'ensemble des vingt classes d'interfaces de OOPS. Nous les listons par leur « Trait », c'est à dire l'alias à renseigner dans un modèle pour le connecter à OOPS. Nous donnons ensuite une description rapide de chaque interface et les méthodes à implémenter. Il s'agit d'une aide pour pouvoir connecter correctement un modèle. L'idéal serait ensuite de compléter cette description, méthode par méthode, afin d'en faire une documentation complète.

Tableau A.1 – Récapitulatif des interfaces d'OOPS et de leurs méthodes

Trait	Description	Méthodes
Covariance	Définit les opérations associés aux matrices de covariance d'erreur de l'ébauche (B) ou du modèle (Q)	<pre>ErrorCovariance(const Geometry &, const Variables &, const eckit::Configuration &); void linearize(const State &, const Geometry &); void multiply(const Increment &, Increment &) const; void inverseMultiply(const Increment &, Increment &) const; void randomize(Increment &) const; std::ostream & operator<< (std::ostream &, const ErrorCovariance &);</pre>
Geometry	Définit la résolution spatiale du modèle	<pre>Geometry(const eckit::Configuration &); Geometry(const Geometry &); std::ostream & operator<< (std::ostream &, const Geometry &);</pre>
Increment	Définit l'incrément δx pour l'assimilation de donnée incrémentale. Correspond également à la différence entre deux états x_1 et x_2	<pre>Increment(const Geometry &, const Variables &, const util::DateTime &); Increment(const Geometry &, const Increment &); Increment(const Increment &, const bool copy); void interpolateTL(const Locations &, ModelAtLocations &) const; void interpolateAD(const Locations &, const ModelAtLocations &); void diff(const State &, const State &); void zero(); void zero(const util::DateTime &); void axpy(const double &, const Increment &, const bool check); double dot_product_with(const Increment &) const; void schur_product_with(const Increment &); void random(); void accumul(const double &, const State &); void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; const util::DateTime validTime() const; void updateTime(const util::Duration &); std::ostream & operator<< (std::ostream &, const Increment &);</pre>
LinearModel	Contient les équations du modèle tangent linéaire (TL) ainsi que du modèle adjoint (AD) du modèle direct	<pre>LinearModel(const Geometry &, const eckit::Configuration &); void setTrajectory(const State &, const ModelAux &); void initializeTL(Increment &) const; void stepTL(Increment &, const ModelAuxIncr &) const; void finalizeTL(Increment &) const; void initializeAD(Increment &) const;</pre>

Voir suite page suivante

Tableau A.1 – suite

Trait	Description	Méthodes
		<pre>void stepAD(Increment &, const ModelAuxIncr &) const; void finalizeAD(Increment &) const; std::ostream & operator<< (std::ostream &, const LinearModel &); static LinearObsOperator * create(const ObsOperator &); LinearObsOperator(const LinearObsOperator &); void setTrajectory(const ModelAtLocations &, const ObsAuxControl &); void obsEquivTL(const ModelAtLocations &, ObsVector &, const ObsAuxIncrement &) const; void obsEquivAD(ModelAtLocations &, const ObsVector &, ObsAuxIncrement &) const; Variables variables() const; std::ostream & operator<< (std::ostream &, const LinearObsOperator &); Locations(const ObservationSpace & os, const util::DateTime &, const util::DateTime &); Localization(const Geometry &, const eckit::Configuration &); void multiply(Increment &) const; std::ostream & operator<< (std::ostream &, const Localization &); Model(const Geometry &, const eckit::Configuration &); void initialize(State &, ModelAux &) const; void step(State &, ModelAux &) const; void finalize(State &, ModelAux &) const; std::ostream & operator<< (std::ostream &, const Model &); ModelAtLocations(const ObservationSpace &, const Variables &, const util::DateTime &, const util::DateTime &); double dot_product_with(const ModelAtLocations &) const; void zero(); ModelAuxControl(const Geometry &, const eckit::Configuration &); ModelAuxControl(const Geometry &, const ModelAuxControl &); ModelAuxControl(const ModelAuxControl &, const bool copy); void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; ModelAuxControl & operator+=(const ModelAuxIncrement &); std::ostream & operator<< (std::ostream &, const ModelAuxControl &); ModelAuxCovariance(const eckit::Configuration &, const Geometry &); void linearize(const ModelAuxControl &, const Geometry &); void multiply(const ModelAuxIncrement &, ModelAuxIncrement &) const; void inverseMultiply(const ModelAuxIncrement &, ModelAuxIncrement &) const;</pre>
LinearObsOperator	Applique l'opérateur d'observation pour le modèle tangent linéaire et adjoint	
Locations	Définit le domaine des observations	
LocalizationMatrix	Définit la matrice de covariance d'erreur pour l'assimilation d'ensemble	
Model	Contient les équations du modèle direct	
ModelAtLocations	Permet de définir le passage du domaine du modèle à celui des observations (H)	
ModelAuxControl	Définit un modèle auxiliaire (biais, flux extérieur, etc)	
ModelAuxCovariance	Matrice de covariance d'erreur du modèle auxiliaire	

Voir suite page suivante

Tableau A.1 – suite

Trait	Description	Méthodes
ModelAuxIncrement	Incrément associé au modèle auxiliaire	<pre>void randomize(ModelAuxIncrement &) const; const eckit::Configuration & config(); std::ostream & operator<< (std::ostream &, const ModelAuxCovariance &); ModelAuxIncrement(const Geometry &, const eckit::Configuration &); ModelAuxIncrement(const ModelAuxIncrement &, const bool copy); ModelAuxIncrement(const ModelAuxIncrement &, const eckit::Configuration &); void diff(const ModelAuxControl &, const ModelAuxControl &); void zero(); ModelAuxIncrement & operator=(const ModelAuxIncrement &); ModelAuxIncrement & operator+=(const ModelAuxIncrement &); ModelAuxIncrement & operator-=(const ModelAuxIncrement &); ModelAuxIncrement & operator*=(const double &); void axpy(const double &, const ModelAuxIncrement &); double dot_product_with(const ModelAuxIncrement &) const; void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; std::ostream & operator<< (std::ostream &, const ModelAuxIncrement &); ObsAuxControl(const eckit::Configuration &); ObsAuxControl(const ObsAuxControl &, const bool copy); void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; ObsAuxControl & operator+=(const ObsAuxIncrement &); std::ostream & operator<< (std::ostream &, const ObsAuxControl &); ObsAuxCovariance(const eckit::Configuration &); void linearize(const ObsAuxControl_ &); void multiply(const ObsAuxIncrement &, ObsAuxIncrement &) const; void inverseMultiply(const ObsAuxIncrement &, ObsAuxIncrement &) const; void randomize(ObsAuxIncrement &) const; const eckit::Configuration & config(); std::ostream & operator<< (std::ostream &, const ObsAuxCovariance &); ObsAuxCovariance(const eckit::Configuration &); ObsAuxIncrement(const eckit::Configuration &); ObsAuxIncrement(const ObsAuxIncrement &, const bool copy);</pre>
ObsAuxControl	Classe pour gérer les observation du modèle auxiliaire	
ObsAuxCovariance	Gère la matrice de covariance d'erreur d'observation du modèle auxiliaire	
ObsAuxIncrement	Incrément d'observation pour le modèle auxiliaire	

Voir suite page suivante

Tableau A.1 – suite

Trait	Description	Méthodes
		<pre> ObsAuxIncrement(const ObsAuxIncrement &, const eckit::Configuration &); void diff(const ObsAuxControl &, const ObsAuxControl &); void zero(); ObsAuxIncrement & operator=(const ObsAuxIncrement &); ObsAuxIncrement & operator+=(const ObsAuxIncrement &); ObsAuxIncrement & operator-=(const ObsAuxIncrement &); ObsAuxIncrement & operator*=(const double &); void axpy(const double &, const ObsAuxIncrement &); double dot_product_with(const ObsAuxIncrement &) const; void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; std::ostream & operator<< (std::ostream &, const ObsAuxIncrement &); </pre>
ObsSpace	Gère les observations du modèle	<pre> ObservationSpace(const eckit::Configuration &, const util::DateTime &, const util::DateTime &); void generateDistribution(const eckit::Configuration &); const util::DateTime & windowStart(); const util::DateTime & windowEnd(); std::ostream & operator<< (std::ostream &, const ObservationSpace &); </pre>
ObsOperator	Opérateur d'observation du modèle (H)	<pre> static ObsOperator * create(ObservationSpace &, const eckit::Configuration &); void obsEquiv(const ModelAtLocations &, ObsVector &, const ObsAuxControl &) const; Variables variables() const; void generateObsError(const eckit::Configuration &); </pre>
ObsVector	Permet de définir le vecteur d'observation et ses opérations associées	<pre> ObsVector(const ObservationSpace &); ObsVector(const ObsVector &, const bool copy); unsigned int size() const; ObsVector & operator=(const ObsVector &); ObsVector & operator*=(const double &); ObsVector & operator+=(const ObsVector &); ObsVector & operator-=(const ObsVector &); ObsVector & operator*=(const ObsVector &); ObsVector & operator/=(const ObsVector &); void zero(); void axpy(const double &, const ObsVector &); void invert(); </pre>

Voir suite page suivante

Tableau A.1 – suite

Trait	Description	Méthodes
		<pre>void random(); double dot_product_with(const ObsVector &) const; double rms() const; void read(const std::string &); void save(const std::string &) const; std::ostream & operator<< (std::ostream &, const ObsVector &);</pre>
State	Définit le vecteur d'état du modèle	<pre>State(const Geometry &, const Variables &, const util::DateTime &); State(const Geometry &, const eckit::Configuration &); State(const Geometry &, const State &); State(const State &); State & operator=(const State &); void interpolate(const Locations &, ModelAtLocations &); void read(const eckit::Configuration &); void write(const eckit::Configuration &) const; double norm() const; void zero(); void accumul(const double &, const State &); std::ostream & operator<< (std::ostream &, const State &);</pre>
Variables	Définit des variables supplémentaires du modèle	<pre>Variables(const eckit::Configuration &); std::ostream & operator<< (std::ostream &, const Variables &);</pre>

Bibliographie

- J. L. ANDERSON : An Ensemble Adjustment Kalman Filter for Data Assimilation. *Monthly Weather Review*, 129(12):2884–2903, 2001. URL [http://dx.doi.org/10.1175/1520-0493\(2001\)129<2884:AEAKFF>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(2001)129<2884:AEAKFF>2.0.CO;2).
- D. AUROUX, P. BANSART et J. BLUM : An evolution of the back and forth nudging for geophysical data assimilation : application to burgers equation and comparisons. *Inverse Problems in Science and Engineering*, 21(3):399–419, 2013. URL <https://doi.org/10.1080/17415977.2012.712528>.
- D. AUROUX et J. BLUM : Back and forth nudging algorithm for data assimilation problems. *Comptes Rendus Mathematique*, 340(12):873 – 878, 2005. ISSN 1631-073X. URL <http://www.sciencedirect.com/science/article/pii/S1631073X05002074>.
- J. BALLABRERA-POY, E. KALNAY et S.-C. YANG : Data assimilation in a system with two scales—combining two initialization techniques. *Tellus A*, 61(4), 2009. ISSN 1600-0870. URL <http://dx.doi.org/10.3402/tellusa.v61i4.15608>.
- R. N. BANNISTER : A review of forecast error covariance statistics in atmospheric variational data assimilation. I : Characteristics and measurements of forecast error covariances. *Quarterly Journal of the Royal Meteorological Society*, 134(637):1951–1970, 2008a. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.339>.
- R. N. BANNISTER : A review of forecast error covariance statistics in atmospheric variational data assimilation. II : Modelling the forecast error covariance statistics. *Quarterly Journal of the Royal Meteorological Society*, 134(637):1971–1996, 2008b. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.340>.
- C. H. BISHOP, B. J. ETHERTON et S. J. MAJUMDAR : Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I : Theoretical Aspects. *Monthly Weather Review*, 129(3):420–436, 2001. URL [http://dx.doi.org/10.1175/1520-0493\(2001\)129<0420:ASWTET>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(2001)129<0420:ASWTET>2.0.CO;2).
- M. A. CANE, A. KAPLAN, R. N. MILLER, B. TANG, E. C. HACKERT et A. J. BUSALACCHI : Mapping tropical Pacific sea level : Data assimilation via a reduced state space Kalman filter. *Journal of Geophysical Research : Oceans*, 101(C10):22599–22617, 1996. ISSN 2156-2202. URL <http://dx.doi.org/10.1029/96JC01684>.

- M. A. CANE, S. E. ZEBIAK et S. C. DOLAN : Experimental forecasts of El Niño. *Nature*, 321:827–832, 1986. URL <http://dx.doi.org/10.1038/321827a0>.
- D. CHAPELLE, M. FRAGU, V. MALLET et P. MOIREAU : Fundamental principles of data assimilation underlying the Verdandi library : applications to biophysical model personalization within euHeart. *Medical and Biological Engineering and Computing*, 51:1221–1233, 2013. URL <https://hal.inria.fr/hal-00760887>.
- D. CHEN : *Coupled data assimilation for ENSO prediction*, vol. 18, chap. 3, p. 45–62. World Scientific Publishing Company, 2012. URL http://dx.doi.org/10.1142/9789812838148_0003.
- D. CHEN, S. E. ZEBIAK, A. J. BUSALACCHI et M. A. CANE : An Improved Procedure for El Niño Forecasting : Implications for Predictability. *Science*, 269(5231):1699–1702, 1995. ISSN 0036-8075. URL <http://science.sciencemag.org/content/269/5231/1699>.
- D. CHEN, S. E. ZEBIAK, M. A. CANE et A. J. BUSALACCHI : Initialization and Predictability of a Coupled ENSO Forecast Model. *Monthly Weather Review*, 125(5):773–788, 1997. URL [http://dx.doi.org/10.1175/1520-0493\(1997\)125<0773:IAP0AC>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1997)125<0773:IAP0AC>2.0.CO;2).
- P. COURTIER, J.-N. THÉPAUT et A. HOLLINGSWORTH : A strategy for operational implementation of 4D-Var, using an incremental approach. *Q.J.R. Meteorol. Soc*, 120(519):1367–1387, 1994. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.49712051912>.
- J. CUXART, P. BOUGEAULT et J.-L. REDELSPERGER : A turbulence scheme allowing for mesoscale and large-eddy simulations. *Quarterly Journal of the Royal Meteorological Society*, 126(562): 1–30, 2000. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.49712656202>.
- N. DAGET : Revue des méthodes d’assimilation. Rap. tech., CERFACS, 2007.
- J. ECKSTEIN et D. P. BERTSEKAS : On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(1):293–318, Apr 1992. ISSN 1436-4646. URL <https://doi.org/10.1007/BF01581204>.
- G. EVENSEN : Advanced Data Assimilation for Strongly Nonlinear Dynamics. *Monthly Weather Review*, 125(6):1342–1354, 1997. URL [http://dx.doi.org/10.1175/1520-0493\(1997\)125<1342:ADAFSN>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1997)125<1342:ADAFSN>2.0.CO;2).
- A. FOUILLOUX et A. PIACENTINI : *The PALM Project : MPMD Paradigm for an Oceanic Data Assimilation Software*, p. 1423–1430. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999. ISBN 978-3-540-48311-3. URL http://dx.doi.org/10.1007/3-540-48311-X_200.
- A. M. FOWLER et A. S. LAWLESS : An idealized study of coupled atmosphere–ocean 4d-var in the presence of model error. *Monthly Weather Review*, 144(10):4007–4030, 2016. URL <https://doi.org/10.1175/MWR-D-15-0420.1>.
- S. FROLOV, C. H. BISHOP, T. HOLT, J. CUMMINGS et D. KUHL : Facilitating strongly coupled ocean–atmosphere data assimilation with an interface solver. *Monthly Weather Review*, 144(1):3–20, 2016. URL <https://doi.org/10.1175/MWR-D-15-0041.1>.

- D. GABAY : Chapter ix applications of the method of multipliers to variational inequalities. In M. FORTIN et R. GLOWINSKI, édés : *Augmented Lagrangian Methods : Applications to the Numerical Solution of Boundary-Value Problems*, vol. 15 de *Studies in Mathematics and Its Applications*, p. 299 – 331. Elsevier, 1983. URL <http://www.sciencedirect.com/science/article/pii/S0168202408700341>.
- D. GABAY et B. MERCIER : A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17 – 40, 1976. ISSN 0898-1221. URL <http://www.sciencedirect.com/science/article/pii/0898122176900031>.
- E. GALANTI, E. TZIPERMAN, M. HARRISON, A. ROSATI et Z. SIRKES : A Study of ENSO Prediction Using a Hybrid Coupled Model and the Adjoint Method for Data Assimilation. *Monthly Weather Review*, 131(11):2748–2764, 2003. URL [http://dx.doi.org/10.1175/1520-0493\(2003\)131<2748:ASOEPU>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(2003)131<2748:ASOEPU>2.0.CO;2).
- M. J. GANDER et L. HALPERN : Optimized Schwarz Waveform Relaxation Methods for Advection Reaction Diffusion Problems. *SIAM Journal on Numerical Analysis*, 45(2):666–697, 2007. URL <http://dx.doi.org/10.1137/050642137>.
- I. Y. GEJADZE et J. MONNIER : On a 2D ‘zoom’ for the 1D shallow water model : Coupling and data assimilation. *Comput. Methods Appl. Mech. and Engrg.*, 196(45–48):4628 – 4643, 2007. ISSN 0045-7825.
- R. GLOWINSKI et A. MARROCO : Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de dirichlet non linéaires. *ESAIM : Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, 9(R2):41–76, 1975. URL <http://eudml.org/doc/193269>.
- S. GRATTON, A. S. LAWLESS et N. K. NICHOLS : Approximate gauss–newton methods for nonlinear least squares problems. *SIAM Journal on Optimization*, 18(1):106–132, 2007. URL <https://doi.org/10.1137/050624935>.
- A. K. GRIFFITH et N. K. NICHOLS : Adjoint Methods in Data Assimilation for Estimating Model Error. *Flow, Turbulence and Combustion*, 65(3):469–488, Dec 2000. ISSN 1573-1987. URL <https://doi.org/10.1023/A:1011454109203>.
- L. HASCOËT et V. PASCUAL : The Tapenade Automatic Differentiation tool : Principles, Model, and Specification. *ACM Transactions On Mathematical Software*, 39(3), 2013. URL <http://dx.doi.org/10.1145/2450153.2450158>.
- K. IDE, P. COURTIER, M. GHIL et A. C. LORENC : Unified Notation for Data Assimilation. *Journal of the Meteorological Society of Japan*, 75(1B):181–189, 1997.
- A. JAZWINSKI : *Stochastic Processes and Filtering Theory*. Mathematics in Science and Engineering. Elsevier Science, 1970. ISBN 9780080960906. URL <https://books.google.fr/books?id=nG1SNvKyY2MC>.
- R. E. KALMAN : A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, 1960. URL <http://dx.doi.org/10.1115/1.3662552>.

- R. E. KALMAN et R. S. BUCY : New Results in Linear Filtering and Prediction Theory. *Journal of Basic Engineering*, 83(1):95–108, 1961. URL <http://dx.doi.org/10.1115/1.3658902>.
- D. E. KEYES, L. C. MCINNES, C. WOODWARD, W. GROPP, E. MYRA, M. PERNICE, J. BELL, J. BROWN, A. CLO, J. CONNORS, E. CONSTANTINESCU, D. ESTEP, K. EVANS, C. FARHAT, A. HAKIM, G. HAMMOND, G. HANSEN, J. HILL, T. ISAAC, X. JIAO, K. JORDAN, D. KAUSHIK, E. KAXIRAS, A. KONIGES, K. LEE, A. LOTT, Q. LU, J. MAGERLEIN, R. MAXWELL, M. MCCOURT, M. MEHL, R. PAWLOWSKI, A. P. RANDLES, D. REYNOLDS, B. RIVIÈRE, U. RÜDE, T. SCHEIBE, J. SHADID, B. SHEEHAN, M. SHEPHARD, A. SIEGEL, B. SMITH, X. TANG, C. WILSON et B. WOHLMUTH : Multiphysics simulations : Challenges and opportunities. *The International Journal of High Performance Computing Applications*, 27(1):4–83, 2013. URL <https://doi.org/10.1177/1094342012468181>.
- D. KONDRASHOV, C. SUN et M. GHIL : Data Assimilation for a Coupled Ocean–Atmosphere Model. Part II : Parameter Estimation. *Monthly Weather Review*, 136(12):5062–5076, 2008. URL <http://dx.doi.org/10.1175/2008MWR2544.1>.
- S. LAKSHMIVARAHAN et J. M. LEWIS : *Nudging Methods : A Critical Overview*, p. 27–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. ISBN 978-3-642-35088-7. URL http://dx.doi.org/10.1007/978-3-642-35088-7_2.
- P. LALOY AUX : The ECMWF coupled assimilation system for climate reanalysis. URL http://www.meteo.fr/cic/meetings/2016/CDAW2016/presentations/1_3.pdf. International workshop on coupled data assimilation (Toulouse, France), 2016.
- P. LALOY AUX, M. BALMASEDA, D. DEE, K. MOGENSEN et P. JANSSEN : A coupled data assimilation system for climate reanalysis. *Q.J.R. Meteorol. Soc.*, p. n/a–n/a, 2015. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.2629>.
- W. G. LARGE : *An Integrated View of Oceanography : Ocean Weather Forecasting in the 21st Century*. Kluwer, 2006a.
- W. G. LARGE, J. C. MCWILLIAMS et S. C. DONEY : Oceanic vertical mixing : A review and a model with a nonlocal boundary layer parameterization. *Reviews of Geophysics*, 32(4):363–403, 1994. ISSN 1944-9208. URL <http://dx.doi.org/10.1029/94RG01872>.
- W. B. LARGE : *Surface Fluxes for Practitioners of Global Ocean Data Assimilation*, p. 229–270. Springer Netherlands, Dordrecht, 2006b. ISBN 978-1-4020-4028-3. URL https://doi.org/10.1007/1-4020-4028-8_9.
- M. LATIF, T. P. BARNETT, M. A. CANE, M. FLÜGEL, N. E. GRAHAM, H. von STORCH, J. S. XU et S. E. ZEBIAK : A review of ENSO prediction studies. *Climate Dynamics*, 9(4):167–179, 1994. ISSN 1432-0894. URL <http://dx.doi.org/10.1007/BF00208250>.
- F. X. LE DIMET et O. TALAGRAND : Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus A*, 38A(2):97–110, 1986. ISSN 1600-0870. URL <http://dx.doi.org/10.1111/j.1600-0870.1986.tb00459.x>.
- T. LEE, J.-P. BOULANGER, A. FOO, L.-L. FU et R. GIERING : Data assimilation by an intermediate coupled ocean-atmosphere model : Application to the 1997–1998 El Niño. *Journal of Geophysical Research : Oceans*, 105(C11):26063–26087, 2000. ISSN 2156-2202. URL <http://dx.doi.org/10.1029/2000JC900118>.

- F. LEMARIÉ, E. BLAYO et L. DEBREU : Analysis of Ocean-atmosphere Coupling Algorithms : Consistency and Stability. *Procedia Computer Science*, 51:2066 – 2075, 2015. ISSN 1877-0509. URL <http://www.sciencedirect.com/science/article/pii/S1877050915012818>.
- F. LEMARIÉ, L. DEBREU et E. BLAYO : Toward an optimized global-in-time Schwarz Algorithm for diffusion equations with discontinuous and spatially variable coefficients. Part 1 : The constant coefficients case. *Electron. Trans. Numer. Anal.*, 40:148–169, 2013a.
- F. LEMARIÉ, L. DEBREU et E. BLAYO : Toward an optimized global-in-time Schwarz Algorithm for diffusion equations with discontinuous and spatially variable coefficients. Part 2 : The variable coefficients case. *Electron. Trans. Numer. Anal.*, 40:170–186, 2013b.
- F. LEMARIÉ, P. MARCHESIELLO, L. DEBREU et E. BLAYO : Sensitivity of ocean-atmosphere coupled models to the coupling method : example of tropical cyclone Erica. *INRIA*, Research report RR-8651, 2014. URL <https://hal.inria.fr/hal-00872496/PDF/RR-8651.pdf>.
- J. L. LIONS, R. TEMAM et S. WANG : New formulations of the primitive equations of atmosphere and applications. *Nonlinearity*, 5(2):237, 1992a. URL <http://stacks.iop.org/0951-7715/5/i=2/a=001>.
- J. L. LIONS, R. TEMAM et S. WANG : On the equations of the large-scale ocean. *Nonlinearity*, 5(5):1007, 1992b. URL <http://stacks.iop.org/0951-7715/5/i=5/a=002>.
- C. LIU, Q. XIAO et B. WANG : An ensemble-based four-dimensional variational data assimilation scheme. part i : Technical formulation and preliminary test. *Monthly Weather Review*, 136(9):3363–3373, 2008. URL <https://doi.org/10.1175/2008MWR2312.1>.
- C. LIU, Q. XIAO et B. WANG : An ensemble-based four-dimensional variational data assimilation scheme. part ii : Observing system simulation experiments with advanced research wrf (arw). *Monthly Weather Review*, 137(5):1687–1704, 2009. URL <https://doi.org/10.1175/2008MWR2699.1>.
- Z. LIU, S. WU, S. ZHANG, Y. LIU et X. RONG : Ensemble data assimilation in a simple coupled climate model : The role of ocean-atmosphere interaction. *Advances in Atmospheric Sciences*, 30(5):1235–1248, 2013. ISSN 1861-9533. URL <http://dx.doi.org/10.1007/s00376-013-2268-z>.
- A. C. LORENC : Analysis methods for numerical weather prediction. *Quarterly Journal of the Royal Meteorological Society*, 112(474):1177–1194, 1986. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.49711247414>.
- E. N. LORENZ et K. A. EMANUEL : Optimal sites for supplementary weather observations : Simulation with a small model. *Journal of the Atmospheric Sciences*, 55(3):399–414, 1998. URL [http://dx.doi.org/10.1175/1520-0469\(1998\)055<0399:OSFSWO>2.0.CO;2](http://dx.doi.org/10.1175/1520-0469(1998)055<0399:OSFSWO>2.0.CO;2).
- F. LU, Z. LIU, S. ZHANG et Y. LIU : Strongly Coupled Data Assimilation Using Leading Averaged Coupled Covariance (LACC). Part I : Simple Model Study. *Monthly Weather Review*, 143(9):3823–3837, 2015. URL <http://dx.doi.org/10.1175/MWR-D-14-00322.1>.
- X. LUO et I. HOTEIT : Ensemble kalman filtering with a divided state-space strategy for coupled data assimilation problems. *Monthly Weather Review*, 142(12):4542–4558, 2014. URL <https://doi.org/10.1175/MWR-D-13-00402.1>.

- P. LYNCH, R. MCGRATH et A. MCDONALD : Digital filter initialization for hirlam. Rap. tech., HIRLAM Technical Report No. 42, 1999.
- J. MCWILLIAMS : *Fundamentals of Geophysical Fluid Dynamics*. Cambridge University Press, 2006.
- A. MONIN et A. OBUKHOV : Basic laws of turbulent mixing in the surface layer of the atmosphere. *Contrib. Geophys. Inst. Acad. Sci. USSR*, 151(163):e187, 1954.
- D. P. MULHOLLAND, P. LALOY AUX, K. HAINES et M. A. BALMASEDA : Origin and Impact of Initialization Shocks in Coupled Atmosphere-Ocean Forecasts. *Mon. Wea. Rev.*, 143:4631–4644, 2015.
- E. OTT, B. R. HUNT, I. SZUNYOGH, A. V. ZIMIN, E. J. KOSTELICH, M. CORAZZA, E. KALNAY, D. J. PATIL et J. A. YORKE : A local ensemble kalman filter for atmospheric data assimilation. *Tellus A*, 56(5):415–428, 2004. ISSN 1600-0870. URL <http://dx.doi.org/10.1111/j.1600-0870.2004.00076.x>.
- C. PELLETIER, F. LEMARIÉ et E. BLAYO : Sensitivity analysis and metamodels for the bulk parameterization of turbulent air-sea fluxes. *Q.J.R. Meteorol. Soc.*, in revision, 2017.
- S. G. PENNY, S. AKELLA, O. ALVES, C. BISHOP, M. BUEHNER, M. CHEVALLIER, F. COUNILLON, C. DRAPER, S. FROLOV, Y. FUJII, A. KARSPECK, A. KUMAR, P. LALOY AUX, J.-F. MAHFOUF, M. MARTIN, M. PEÑA, P. de ROSNAY, A. SUBRAMANIAN, R. TARDIF, Y. WANG et X. WU : Coupled data assimilation for integrated earth system analysis and prediction : Goals, challenges and recommendations. Rap. tech. 3, World Meteorological Organization, World Weather Research Programme, International workshop on coupled data assimilation (18-21 Octobre 2016, Toulouse), 2017. URL https://www.wmo.int/pages/prog/arep/wwrp/new/documents/Final_WWRP_2017_3_27_July.pdf.
- J. PULLEN, J. D. DOYLE et R. P. SIGNELL : Two-Way Air-Sea Coupling : A Study of the Adriatic. *Mon. Wea. Rev.*, 134(5):1465–1483, 2006. URL <http://dx.doi.org/10.1175/MWR3137.1>.
- A. M. QUARTERONI, R. SACCO et F. SALERI : *Méthodes Numériques*. Springer-Verlag Mailand, 2007.
- A. ROSATI, K. MIYAKODA et R. GUDGEL : The Impact of Ocean Initial Conditions on ENSO Forecasting with a Coupled Model. *Monthly Weather Review*, 125(5):754–772, 1997. URL [http://dx.doi.org/10.1175/1520-0493\(1997\)125<0754:TI00IC>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1997)125<0754:TI00IC>2.0.CO;2).
- P. SMITH, A. FOWLER et A. LAWLESS : Exploring strategies for coupled 4D-Var data assimilation using an idealised atmosphere-ocean model. *Tellus A*, 67(0), 2015. ISSN 1600-0870. URL <http://dx.doi.org/10.3402/tellusa.v67.27025>.
- P. J. SMITH, A. S. LAWLESS et N. K. NICHOLS : Estimating forecast error covariances for strongly coupled atmosphere-ocean 4d-var data assimilation. *Monthly Weather Review*, 145(10):4011–4035, 2017. URL <https://doi.org/10.1175/MWR-D-16-0284.1>.
- R. B. STULL, éd. *An Introduction to Boundary Layer Meteorology*. Springer Netherlands, 1988. ISBN 978-90-277-2769-5.

- N. SUGIURA, T. AWAJI, S. MASUDA, T. MOCHIZUKI, T. TOYODA, T. MIYAMA, H. IGARASHI et Y. ISHIKAWA : Development of a four-dimensional variational coupled data assimilation system for enhanced analysis and prediction of seasonal to interannual climate variations. *Journal of Geophysical Research : Oceans*, 113(C10):n/a–n/a, 2008. ISSN 2156-2202. URL <http://dx.doi.org/10.1029/2008JC004741>. C10017.
- C. SUN, Z. HAO, M. GHIL et J. D. NEELIN : Data Assimilation for a Coupled Ocean–Atmosphere Model. Part I : Sequential State Estimation. *Mon. Wea. Rev.*, 130:1073–1099, 2002.
- R. TARDIF, G. J. HAKIM et C. SNYDER : Coupled atmosphere–ocean data assimilation experiments with a low-order climate model. *Climate Dynamics*, 43(5):1631–1643, 2013. ISSN 1432-0894. URL <http://dx.doi.org/10.1007/s00382-013-1989-0>.
- Y. TRÉMOLET : Accounting for an imperfect model in 4d-var. *Quarterly Journal of the Royal Meteorological Society*, 132(621):2483–2504, 2006. ISSN 1477-870X. URL <http://dx.doi.org/10.1256/qj.05.224>.
- Y. TRÉMOLET : Incremental 4d-var convergence study. *Tellus A*, 59(5):706–718, 2007. ISSN 1600-0870. URL <http://dx.doi.org/10.1111/j.1600-0870.2007.00271.x>.
- Y. TRÉMOLET : Software Infrastructure for Numerical Weather Prediction, OOPS Project. Séminaire JCSDA, Conference Center, NOAA Center for Weather and Climate Prediction, 5830 University Research Court, College Park, MD, 2015. URL <https://www.jcsda.noaa.gov/documents/seminardocs/2015/Tremolet20150608.pdf>.
- S. VALCKE, T. CRAIG et L. COQUART : OASIS3-MCT User Guide, OASIS3-MCT 3.0. Rap. tech. TR/CMGC/15/38, CERFACS/CNRS SUC URA No 1875, Toulouse, France, 2015. URL http://www.cerfacs.fr/oa4web/oasis3-mct_3.0/oasis3mct_UserGuide.pdf.
- A. VIDARD, E. BLAYO, F.-X. LE DIMET et A. PIACENTINI : 4d variational data analysis with imperfect model. *Flow, Turbulence and Combustion*, 65(3):489–504, Dec 2000. ISSN 1573-1987. URL <https://doi.org/10.1023/A:1011452303647>.
- A. WEAVER et P. COURTIER : Correlation modelling on the sphere using a generalized diffusion equation. *Quarterly Journal of the Royal Meteorological Society*, 127(575):1815–1846, 2001. ISSN 1477-870X. URL <http://dx.doi.org/10.1002/qj.49712757518>.
- S. E. ZEBIAK et M. A. CANE : A Model El Niño–Southern Oscillation. *Monthly Weather Review*, 115(10):2262–2278, 1987. URL [http://dx.doi.org/10.1175/1520-0493\(1987\)115<2262:AMENO>2.0.CO;2](http://dx.doi.org/10.1175/1520-0493(1987)115<2262:AMENO>2.0.CO;2).
- S. ZHANG : A Study of Impacts of Coupled Model Initial Shocks and State–Parameter Optimization on Climate Predictions Using a Simple Pycnocline Prediction Model. *Journal of Climate*, 24(23):6210–6226, 2011. URL <http://dx.doi.org/10.1175/JCLI-D-10-05003.1>.
- S. ZHANG, M. J. HARRISON, A. ROSATI et A. WITTENBERG : System Design and Evaluation of Coupled Ensemble Data Assimilation for Global Oceanic Climate Studies. *Monthly Weather Review*, 135(10):3541–3564, 2007. URL <http://dx.doi.org/10.1175/MWR3466.1>.