

14^e édition des Journées francophones

EGC 2014

Extraction et Gestion des Connaissances

28-31 janvier

RENNES

IRISA & Centre Inria Rennes - Bretagne Atlantique
Campus de Beaulieu, Rennes

Journées Ateliers/Tutoriels

Couplage : Couplage des sources d'information hétérogènes

Atelier “Assimilation de données complexes et hétérogènes”

Organisateurs :

- Emmanuel Frenod
 - LMBA, Laboratoire de Mathématiques de Bretagne Atlantique, UMR CNRS 6205, Université de Bretagne-Sud
- Thomas Corpetti
 - OSUR, Observatoire des Sciences de l’Univers de Rennes, UMS 3343, CNRS
- François-Xavier Le Dimet
 - MOISE, Modélisation, Observation et Identification pour les Sciences de l’Environnement, Laboratoire Jean-Kuntzman, Université de Grenoble

PRÉFACE

Les techniques d'observation de phénomènes physiques mettent en jeu un panel de plus en plus vaste de données spatio-temporelles avec des modalités variables. La manipulation et l'analyse de ces données complexes et hétérogènes est alors un problème délicat qui mérite une attention particulière.

Dans cet atelier, nous nous focalisons sur les techniques d'assimilation de données qui permettent de combiner de manière optimale des opérateurs d'observations avancés avec des modèles physiques sur les phénomènes observés. Un certain nombre de résultats récents sont présentés pour diverses applications (électrocardiologie, assimilation d'images, techniques de différentiation automatique).

Au delà de l'intérêt scientifique, l'objectif de cet atelier est également d'aller vers la structuration d'un groupe autour de ce thème dans le but de préparer des projets de recherche plus ambitieux (par exemple ANR ou européen).

Emmanuel FRENOD Thomas CORPETTI François-Xavier LE DIMET
LMBA OSUR MOISE

Membres du comité de lecture

Le Comité de Lecture est constitué de:

Emmanuel Frenod
Thomas Corpetti

François-Xavier Le Dimet

TABLE DES MATIÈRES

Assimilation de séquences d'images : vers une prise en compte des corrélations spatiales au sein des erreurs d'observation <i>Vincent Chabot, Maëlle Nodet, Nicolas Papadakis, Arthur Vidard</i>	1
Adjoint Automatic Differentiation for Data Assimilation <i>Laurent Hascoët</i>	13
Prise en compte de structures spatiales pour l'assimilation variationnelle de données de télédétection. Exemple sur un modèle simple de croissance de végétation. <i>Claire Lauvernet, François-Xavier Le Dimet, Frédéric Baret</i>	27
Assimilation de données hétérogènes pour des systèmes faiblement couplés. Application au problème inverse de l'électrocardiologie <i>Philippe Moireau, Cesare Corrado, Jean-Frédéric Gerbeau</i>	41
Index des auteurs	53

Assimilation de séquences d'images : vers une prise en compte des corrélations spatiales au sein des erreurs d'observation

Vincent Chabot *, Maëlle Nodet*, Nicolas Papadakis **, Arthur Vidard *

* INRIA, Université Joseph Fourier Grenoble 1,
Laboratoire Jean Kuntzmann 51 rue des Mathématiques, 38041 Grenoble cedex 09
vincent.chabot@imag.fr, maelle.nodet@inria.fr, arthur.vidard@imag.fr

** Institut de Mathématiques de Bordeaux
351 Cours de la Libération, 33405 TALENCE Cedex, France
nicolas.papadakis@math.u-bordeaux1.fr

Résumé. Les images satellites sont une source importante d'information sur l'évolution du système terre. Bien que ces séquences d'images soient de plus en plus utilisées, l'importance des corrélations spatiales entre les erreurs présentes en leur sein est rarement prise en compte en pratique. Cela conduit à une sous utilisation de l'information contenue dans ces données. Dans cet article, une nouvelle manière (peu coûteuse) d'intégrer cette information dans le cadre de l'assimilation de données est proposée. Le problème de l'utilisation d'images corrompues par un bruit fortement corrélé en espace afin de contrôler l'état initial du système est abordé. La faisabilité et la pertinence de l'approche proposée est démontrée dans le cadre d'une configuration académique.

1 Introduction

Un des problèmes rencontrés en prévision numérique du temps est la détermination de l'état initial du système étudié. En effet, les véritables états de l'atmosphère et de l'océan, à un instant donné et en tous points de l'espace, ne sont pas accessibles. Les erreurs sur les conditions initiales peuvent se propager dans le temps et s'amplifier avec l'échéance de la prévision. Pour réduire ces erreurs, en se rapprochant au maximum de l'état réel de l'atmosphère, les informations disponibles, comprenant les observations, une ébauche, les équations du modèle ainsi que des statistiques d'erreurs sur ces trois types d'informations, sont combinées par les schémas d'assimilation de données.

Depuis la fin des années 70, différents satellites ont été mis en orbite dans le but d'accroître notre connaissance de l'atmosphère et de l'océan par l'observation de la Terre. Les satellites géostationnaires produisent, entre autre, des séquences d'images montrant l'évolution dynamique de certains types d'objets météorologiques ou océaniques tels les nuages, les fronts, les vortex, . . .

Actuellement, dans les modèles numériques de prévision du temps, l'information contenue dans de ce type de séquences est assimilée au travers des AMV (Atmospheric Motion Vector).

Le principe est d'estimer le mouvement de certains objets identifiés d'une image à une autre en utilisant des techniques de corrélations croisées puis d'utiliser l'estimation de ce mouvement en tant qu'observation. D'autres méthodes, basées sur l'équation de contrainte du flot optique, visant à estimer des champs de vitesses denses ont été introduites par Papadakis et Mémin (2007).

Pendant ces approches, extrayant des champs de vitesses à partir des séquences d'images puis les utilisant comme observations, souffrent de la difficulté à modéliser les statistiques d'erreur d'observation. L'erreur due au pré-traitement des images ne peut généralement pas être quantifiée de manière précise et l'information contenue dans les champs de vitesses estimés est donc difficile à utiliser en pratique.

Pour contourner ce problème, des méthodes utilisant une comparaison directe entre la variation temporelle de l'image et le champ de vitesses du modèle ont été étudiées notamment dans Papadakis et Mémin (2008) via l'introduction d'un opérateur d'observation non linéaire basé sur l'équation de contrainte du flot optique. Cette approche permet de contourner le problème de la régularisation artificielle en ne considérant que l'information photométrique.

Une autre approche consistant à comparer une séquence d'images observées à une séquence d'images synthétiques créée à partir des sorties du modèle a été introduite dans Titau et al. (2010) et Souopgui (2010).

Dans cet article, une extension de cette méthode utilisant une comparaison de la séquence observée et de la séquence synthétique dans l'espace des gradients de l'image et des espaces d'ondelettes orthonormales est présentée en section 3. L'intérêt de ces deux espaces (en comparaison avec l'espace des pixels) en présence de bruit d'observation fortement corrélé en espace est étudié dans un contexte académique en section 4.

2 Principe général de l'assimilation variationnelle de données

Soit un modèle dynamique \mathcal{M} décrivant l'évolution de la variable d'état \mathbf{X} :

$$\begin{cases} \partial_t \mathbf{X}(\mathbf{X}_0, \mathbf{x}, t) + \mathcal{M}(\mathbf{X}(\mathbf{X}_0, \mathbf{x}, t)) = 0 \\ \mathbf{X}(\mathbf{X}_0, \mathbf{x}, t_0) = \mathbf{X}_0 \end{cases} \quad (1)$$

Soient des observations $Y(t)$ de ces variables d'état. Le but de l'assimilation de données est de déterminer une condition initiale que l'on appelle l'état analysé \mathbf{X}_0^a tel que celui-ci ne soit pas trop éloigné d'une ébauche \mathbf{X}_0^b fournie par une prévision précédente et que la trajectoire issue de \mathbf{X}_0^a soit proche des observations $Y(t)$. Pour cela, \mathbf{X}_0^a est défini comme le minimum de la fonction coût suivante :

$$\begin{aligned} J(\mathbf{X}_0) &= J_b(\mathbf{X}_0) + J_o(\mathbf{X}_0) \\ &= \frac{1}{2} \|\mathbf{X}_0 - \mathbf{X}_0^b\|_{\mathcal{V}}^2 + \frac{1}{2} \int_{t_0}^{t_f} \|Y(t) - \mathcal{H}(\mathbf{X}(\mathbf{X}_0, \mathbf{x}, t))\|_{\mathcal{O}}^2 dt, \end{aligned} \quad (2)$$

où \mathcal{V} est l'espace d'état, \mathcal{O} l'espace d'observation et $\mathcal{H} : \mathcal{V} \mapsto \mathcal{O}$ est appelé l'opérateur d'observation. Pour des problèmes réalistes en prévision numérique du temps la dimension de l'espace d'état est importante ($\sim 10^7 - 10^9$ variables).

Généralement, en assimilation variationnelle, la minimisation est faite par méthode de descente du gradient, celui-ci étant calculé par méthode adjointe Le Dimet et Talagrand (1986).

3 Opérateur d'observation et distances spécifiques aux séquences d'images

L'assimilation de séquences d'images requiert la définition d'une distance entre les images observées et les sorties du modèle. Il a été choisi, comme dans Titaud et al. (2010), de créer un équivalent modèle, noté q^{synth} , de l'image observée, notée q^{obs} , puis de calculer la distance entre la séquence observée et la séquence synthétique.

Pour créer cette séquence d'images synthétiques l'hypothèse que les quantités observées $q^{obs}(t)$ correspondent à des traceurs passifs transportés par la dynamique du système est émise. L'équation d'advection-diffusion utilisée est :

$$\partial_t q + u\partial_x q + v\partial_y q - \nu_T \Delta q = 0 \quad (3)$$

où u et v sont les composantes du champ de vitesses appartenant à l'espace d'état du modèle \mathcal{M} et ν_T est le coefficient de diffusion. Cette équation (3), sous réserve de la connaissance de $q^{synth}(t_0)$, permet de générer une séquence d'images $q^{synth}(t_i)$ à comparer aux images observées $q^{obs}(t_i)$.

Pour cela différentes notions de distance sont utilisées :

- La comparaison pixels à pixels :

$$\begin{aligned} J_o(\mathbf{X}_0) &= \frac{1}{2} \int_{t_0}^{t_f} \|Y(t) - \mathcal{H}(\mathbf{X}(\mathbf{X}_0, \mathbf{x}, t))\|_{\mathcal{O}}^2 \\ &= \sum_{t_0}^{t_f} \|q^{obs}(t) - q^{synth}(t)\|_{R_{Pix}}^2 \end{aligned} \quad (4)$$

avec

$$\|\mathbf{X}\|_R^2 = \mathbf{X}^T R^{-1} \mathbf{X} \quad (5)$$

où R est la matrice de covariance d'erreur d'observation dans l'espace considéré.

L'équation (5) met en évidence le rôle crucial joué par la matrice R dans la définition du problème (2).

Remarque 1. *En pratique, la taille de la matrice R rend son stockage et son inversion inenvisageable. Par conséquent, dans cette étude seul des approximations diagonales de ces matrices de covariances sont utilisées (et ce quel que soit l'espace de travail considéré).*

- La comparaison entre les gradients des images :

$$J_o(\mathbf{X}_0) = \sum_{t_0}^{t_f} \|\nabla q^{obs}(t) - \nabla q^{synth}(t)\|_{R_{Grad}}^2 \quad (6)$$

Assimilation de données images

avec ∇ l'opérateur de gradient spatial calculé en utilisant un schéma de différences finies centré.

Cette distance prend en compte le fait que l'information sur la dynamique du système, au sein de la séquence de traceurs, est principalement portée par les discontinuités.

- Dans des espaces d'ondelettes deux notions de distance différant par l'information utilisée, définies en équations (7) et (9), sont considérées.

La première inclut toute l'information disponible au sein du jeu de coefficients et s'écrit :

$$J_o(\mathbf{X}_0) = \sum_{t_0}^{t_f} \|W(q^{synth}(t)) - W(q^{obs}(t))\|_{R_W}^2 \quad (7)$$

où W est la transformée en ondelette orthonormale choisie.

Remarque 2. Cette distance est identique à la comparaison pixels à pixels (4) lorsque :

$$R_W = W R_{Pix} W^T. \quad (8)$$

Cependant comme R_W et R_{Pix} sont des approximations diagonales des vraies matrices de covariances, la relation (8) n'est vérifiée, pour des ondelettes orthonormales, que lorsque $R_{Pix} = R_W = \sigma^2 \mathbf{I}_n$.

Dans les autres cas, les distances (4) et (7) ne sont pas identiques car les approximations faites sur les statistiques d'erreur ne sont pas équivalentes.

Lorsqu'on souhaite ne conserver que les coefficients ayant un ratio Signal/Bruit important (afin par exemple d'utiliser uniquement des coefficients porteurs d'information), on peut être amené à travailler dans des sous-espaces d'ondelettes. Ainsi la seconde notion de distance utilisée dans un espace d'ondelettes est alors :

$$J_o(\mathbf{X}_0) = \sum_{t_0}^{t_f} \|\mathcal{T}_I \circ W(q^{synth}(t)) - \mathcal{T}_I \circ W(q^{obs}(t))\|_{R_{\mathcal{T}_I \circ W}}^2 \quad (9)$$

où \mathcal{T}_I est un seuillage appliqué aux coefficients de l'image observée permettant la sélection des coefficients importants de l'image (entre lesquels la distance est mesurée). Dans cette étude, le seuillage \mathcal{T}_I consiste à conserver uniquement l'information contenue dans les coefficients d'ondelettes ayant la norme la plus importante.

4 Résultats obtenus sur des expériences jumelles

4.1 Contexte expérimental

Afin de tester la robustesse des distances proposées au bruit présent dans les observations, des expériences jumelles sont effectuées. Celles-ci consistent à retrouver, dans un contexte où la vérité terrain est connue, un état analysé à partir d'observations synthétiques et d'une ébauche.

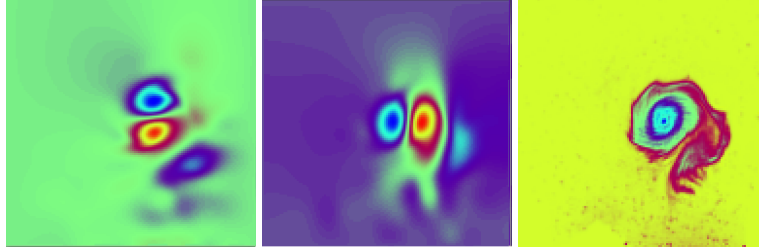


FIG. 1 – A gauche et au centre les "vraies" composantes zonale (u_0^t) et meridionale (v_0^t) du champ de vitesses initial. A droite, la première image du champ de traceur passif (q_1^t).

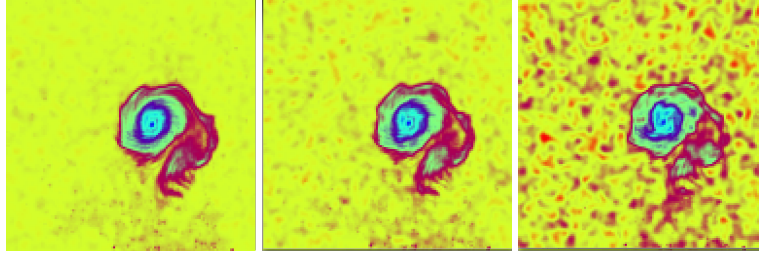


FIG. 2 – Exemples d'observations bruitées pour chaque niveau de bruit étudié. Le ratio Signal/Bruit (SNR) de l'image de gauche est de 26.8 dB, celui de l'image centrale de 20.8 dB et celui de l'image de droite de 14.8 dB.

Dans ces expériences, la dynamique du système est décrite par les équations de Saint-Venant :

$$\begin{cases} \partial_t u - (f + \zeta)v + \partial_x B & = -ru + \kappa \Delta u \\ \partial_t v + (f + \zeta)u + \partial_y B & = -rv + \kappa \Delta v \\ \partial_t h + \partial_x(hu) + \partial_y(hv) & = 0. \end{cases} \quad (10)$$

où $\zeta = \partial_x v - \partial_y u$ est la vorticité relative, $B = g^* h + \frac{u^2 + v^2}{2}$ est le potentiel de Bernoulli pour lequel g^* est la gravité réduite, f est le paramètre de coriolis, κ le coefficient de diffusion et r le coefficient de frottement sur le fond.

La vérité terrain (u_0^t, v_0^t, h_0^t) définie en figure 1 simule l'évolution d'un vortex sur la table tournante CORIOLIS¹. Cette dernière permet de recréer l'effet de la force de Coriolis dans une couche d'eau peu profonde.

Dans les expériences effectuées, seules des observations indirectes (concentration de traceur passif) sont assimilées. Un run direct du modèle utilisant le vrai champ de vitesses permet de générer une séquence d'images q^t à partir de la concentration initiale du traceur passif q_0^t via l'équation (3).

4.2 Génération des observations

Un bruit gaussien corrélé en espace est ajouté à la séquence d'images. Pour une même matrice de corrélation, trois niveaux de bruits sont étudiés. Une image est présentée en figure 2 pour chaque niveau de bruit :

- Sur l'image de gauche, le bruit n'est presque pas visible. Cette séquence d'images est proche des données parfaites présentées en figure 1 ;
- Bien que visible sur la seconde image, le bruit ne modifie pas la perception visuelle des structures principales ;
- La perception visuelle des principales structures est altérée par la présence d'un bruit d'amplitude plus importante sur la troisième image.

4.3 Expériences effectuées

Afin d'évaluer le gain réalisé lors du changement de distance, les expériences jumelles décrites en section 4.1 sont effectuées avec les observations présentées en section 4.2. L'ébauche intervenant dans le problème de minimisation (2) est la suivante : $(u_0^b, v_0^b, h_0^b) = (0, 0, h_{mean})$. Pour chacune des distances définies en section 3, il reste à décrire brièvement la matrice R utilisée. On rappelle qu'on se restreint, dans ces expériences, à n'utiliser que des matrices diagonales.

Matrice R dans l'espace des pixels :

Bien que le bruit soit corrélé en espace, sa répartition au sein des séquences d'images est homogène en temps et en espace. Par conséquent, la variance de chaque pixel est identique. On utilise dans (4) la diagonale de la vraie matrice de covariance (notée R_{Pix}^{true}) qui est :

$$R_{Pix} = \text{Diag}(R_{Pix}^{true}) = \sigma^2 \mathbf{I}_n. \quad (11)$$

Matrice R dans l'espace des gradients :

En plus d'être réparti de manière homogène, le bruit utilisé présente des corrélations isotropes. Par conséquent la diagonale de la vraie matrice de covariance est utilisée dans (6) :

$$R_{Grad} = \text{Diag}(R_{Grad}^{true}) = \tilde{\sigma}^2 \mathbf{I}_{2n}. \quad (12)$$

Remarque 3. La variance de $I_x(i, j)$, composante dans la direction x du gradient, est donnée par :

$$\begin{aligned} \tilde{\sigma}^2 = V(I_x(i, j)) &= V\left(\frac{I(i+1, j) - I(i-1, j)}{2}\right) \\ &= \frac{V(I(i+1, j)) + V(I(i-1, j))}{4} - \frac{\text{Cov}(I(i+1, j), I(i-1, j))}{2} \\ &= \frac{\sigma^2}{2} (1 - \text{Cor}(I(i+1, j), I(i-1, j))) \end{aligned}$$

où $\text{Cor}(I, J)$ est le coefficient de corrélation entre I et J . Cette variance dépend donc fortement des corrélations présentes dans le signal observé. Plus cette corrélation est importante,

1. <http://coriolis.legi.grenoble-inp.fr>

plus la variance de $I_x(i, j)$ diminue (donc moins le bruit a un impact important dans l'espace des gradients). Ainsi, un bruit fortement corrélé est moins fortement répercuté dans l'espace des gradients que dans celui des pixels.

Une autre vision de ceci est de considérer qu'un bruit corrélé en espace présente un biais local auquel les gradients sont moins sensibles par construction.

Matrice R dans un sous-espace d'ondelettes (étude de l'influence du seuillage) :

Dans le but d'étudier l'influence du seuillage sur la minimisation, il a été choisi dans un premier temps de spécifier une matrice de covariance proportionnelle à l'identité dans un sous-espace d'ondelettes :

$$R_{\mathcal{T}_I \circ W} = \sigma^2 \mathbf{I}_k = PW \text{Diag}(R_{Pix}^{true}) W^T P^T \quad (13)$$

où P est la matrice de projection dans un sous-espace d'ondelettes et k est le nombre de fonctions dans ce sous-espace.

Comme spécifié dans la remarque 2, le fait de choisir cette matrices entraine que les normes (4) et (9) sont équivalentes si aucun seuillage \mathcal{T}_I n'est effectué (i-e que $P = \mathbf{I}_n$). La différence entre (4) et (9) réside donc uniquement dans la sélection de l'information à prendre en compte. En comparant les résultats obtenus par ces deux distances, on peut ainsi évaluer l'impact de l'opération de seuillage \mathcal{T}_I sur les performances de la minimisation. Les résultats obtenus avec les ondelettes de Daubechies à 8 moments nuls (notées D_8) sont présentés en section 4.4.

Matrice R dans une base d'ondelettes (étude de l'influence de l'approximation des statistiques d'erreur d'observation) :

Il est possible à partir de la connaissance de la vraie matrice de covariance d'erreur d'observation dans l'espace des pixels de construire la vraie matrice de covariance d'erreur d'observation dans une base d'ondelettes (voir par exemple Vannucci et Corradi (1999)).

Dans la distance définie en équation (7) la diagonale de cette matrice est utilisée :

$$\begin{aligned} R_W &= \text{Diag}(W R_{Pix}^{true} W^T) \\ &\neq W \text{Diag}(R_{Pix}^{true}) W^T. \end{aligned} \quad (14)$$

Cette approximation de la matrice R_W diffère de celle faite en équation (13) par le fait que l'hypothèse diagonale est effectuée après le passage dans la base d'ondelette et non avant.

Pour un bruit corrélé en espace, la variance des coefficients dans une base d'ondelettes varie fortement selon l'échelle à laquelle la fonction d'ondelette associée considère le signal. Par conséquent, à la différence de $R_{\mathcal{T}_I \circ W}$, la matrice R_W utilisée dans (7) est bien non proportionnelle à l'identité. De cette manière une étude de l'influence de l'approximation des vraies statistiques d'erreur est possible en comparant les résultats provenant de l'utilisation des fonctions coûts (4) et (7).

Remarque 4. Dans ces deux cas l'ensemble de l'information (signal et bruit) est utilisé. L'unique différence entre (4) et (7) réside dans l'approximation des statistiques d'erreur d'observation effectuée dont un exemple est présenté en figure 3.

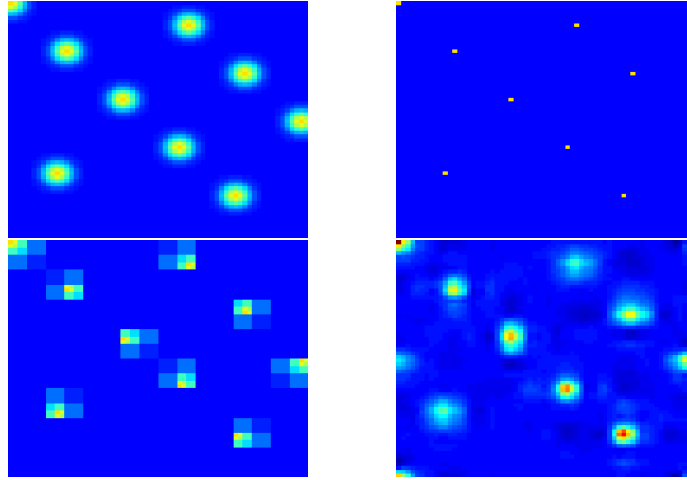


FIG. 3 – En haut à gauche on présente la vraie covariance entre l’erreur en un pixel et l’erreur chez ses voisins pour 10 pixels différents répartis au sein de notre image. En haut à droite est présentée la conséquence de l’approximation diagonale (11) dans la base des pixels. En bas sont présentées les approximations faites lorsque l’hypothèse diagonale introduite en équation (14) est effectuée dans la base de Haar (à gauche) et dans la base de Daubechies à 8 moments nuls (à droite).

La figure 3 présente, pour 10 pixels différents, la distribution spatiale de leur covariance d’erreur d’observation. En haut à gauche, on observe les vraies covariances d’erreur d’observation. Celles-ci sont isotropes et le bruit est réparti de manière homogène dans l’image. L’approximation diagonale dans l’espace des pixels (11) est présentée en haut à droite. Avec cette approximation, on considère que le champ d’erreur d’observation ne présente pas de corrélation spatiale. L’approximation diagonale dans deux bases d’ondelettes différentes (équation (14)), la base de Haar et la base d’ondelettes de Daubechies à 8 moments nuls, est ensuite présentée. Ces deux approximations permettent de tenir compte d’une partie des covariances d’erreur d’observation. On observe que les statistiques d’erreur prescriptibles par une matrice diagonale dans une base d’ondelettes dépendent de la base choisie. L’homogénéité du bruit est conservée uniquement dans la base de Haar tandis que l’isotropie est perdue dans les deux cas.

4.4 Résultats obtenus avec ces différentes distances

Le tableau 1 présente le pourcentage d’erreur résiduelle dans la composante analysée u_0^a par rapport à l’erreur présente dans l’ébauche. Bien que non présentées dans ce document, les performances atteintes pour la composante v sont comparables à celles atteintes pour la composante u . On rappelle que pour toutes les expériences effectuées le champ de vitesses n’est pas observé.

Les résultats obtenus pour la comparaison entre pixels sont présentés en première colonne. On remarque que l’erreur résiduelle dépend fortement du niveau de bruit, ce qui signifie que

	Pix	$\mathcal{T}_I \circ W$	Grad	D8	Haar
14.8 dB	60.8%	60.1%	34.0%	9.3%	22.8%
20.8 dB	26.2%	28.5%	17.8%	7.6%	12.5%
26.8 dB	15.6%	17.1%	12.4%	7.2%	8.4%
Perfect data	7.6%	8.5%	7.4%	7.1%	6.4%

TAB. 1 – Moyenne sur 10 expériences du pourcentage de l’erreur résiduelle dans la composante analysée, u_0^a , par rapport à l’erreur présente dans l’ébauche, u_0^b . Ce pourcentage a été calculé pour différents niveaux de bruit, caractérisés par leur ratio Signal Bruit (SNR), présent au sein des séquences d’images. Dans $\mathcal{T}_I \circ W$ la comparaison s’effectue entre les 20% des coefficients ayant la plus grande norme avec la matrice de covariance définie en équation (13). Dans D8 et Haar, tous les coefficients d’ondelettes sont conservés. Dans ce cas, la matrice de covariance d’erreur est celle de l’équation (14).

la distance entre pixels n’est que peu robuste au bruit. Lorsqu’on compare ces résultats à ceux obtenus dans un sous-espace d’ondelettes avec la matrice $R_{\mathcal{T}_I \circ W}$ (deuxième colonne), on s’aperçoit que le pourcentage d’erreur résiduelle est du même ordre. Ceci peut s’expliquer par le fait que le seuillage des coefficients (\mathcal{T}_I), restreignant le travail à un sous-espace de petite taille mais contenant la majeure partie de l’information, ne change que peu le problème de minimisation (2). Ainsi on peut réduire de manière importante la taille du vecteur d’observation sans pour autant perdre en qualité d’estimation.

La troisième colonne du tableau 1 présente les résultats obtenus lors de la comparaison entre gradients. On observe une plus grande réduction de l’erreur dans ce cas que pour les deux distances précédentes. Ces résultats peuvent s’expliquer si l’on considère que, « localement », un bruit additif corrélé en espace introduit un « biais » dans l’espace des pixels. Les gradients, éliminant par construction une partie de ce biais, sont moins sensibles aux corrélations présentes au sein du bruit.

Les performances atteintes en utilisant une matrice non proportionnelle à l’identité avec les ondelettes de Daubechies à 8 moments nuls et les ondelettes de Haar sont présentées en 4ème et 5ème colonnes du tableau 1. Pour ces expériences, l’ensemble des coefficients d’ondelettes est assimilé. L’erreur résiduelle est dans ces deux cas bien plus faible qu’avec les trois distances précédentes. La robustesse au bruit de la distance utilisant les ondelettes de Daubechies est impressionnante : les erreurs commises lors de l’utilisation des séquences les plus bruitées sont moins importantes que celles commises avec des séquences moins bruitées pour les autres distances à l’exception de celle utilisant les ondelettes de Haar. Ces meilleurs résultats sont dus à l’utilisation d’une matrice non proportionnelle à l’identité. Faire varier la variance de chaque coefficient permet de représenter une partie des corrélations présentes au sein des erreurs du signal original. Les résultats ne sont pas les mêmes pour différentes bases d’ondelettes car, comme présenté en figure 3, les corrélations représentables par une matrice R diagonale varient fortement en fonction de la base choisie.

Une fois ces premiers chiffres analysés se pose la question concernant la localisation des erreurs dans le champ analysé. Les colonnes de la figure 4 permettent de visualiser les champs

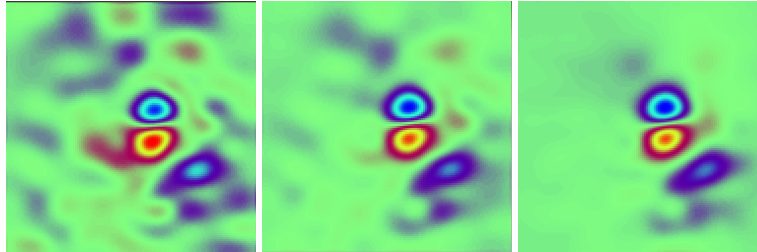


FIG. 4 – Composante u du champ de vitesse analysé pour une séquence d’observation correspondant au bruit fort (SNR 14.8 dB). A gauche est présenté le résultat d’une assimilation basée sur la distance entre pixel, au centre sur la distance entre gradients, à droite sur la distance entre les coefficients d’ondelettes de Daubechies (avec la matrice de covariance définie en (14)).

analysés utilisant la distance entre pixels (à gauche), entre gradients (au milieu) et entre les coefficients d’ondelettes de Daubechies (à droite). Ces figures montrent que le champ de vitesse est globalement bien reconstruit sous le tourbillon quelque soit le cas. Les principales erreurs pour les distances Pixels et Gradients se situent dans les zones lisses des observation où l’absence de discontinuités dans la séquence d’images originale signifie l’absence d’information sur le mouvement sous-jacent. La répercussion de l’erreur d’observation est plus importante pour la comparaison pixels à pixels que pour la comparaison entre gradients. Dans un espace de type ondelettes, il est possible d’intégrer une partie des corrélations du bruit dans la diagonale de la matrice de covariance d’erreur d’observation. La troisième colonne nous montre que cette opération permet de réduire drastiquement l’impact sur le champ de vitesses analysé du bruit présent dans les zones lisses de l’image.

5 Conclusion

Dans ce document, nous introduisons de nouvelles manières d’utiliser l’information présente dans des séquences d’images de traceurs, révélant de manière indirecte l’évolution d’un système géophysique. Les différents résultats obtenus sont encourageants et montrent qu’il est possible d’extraire suffisamment d’information de ces séquences d’images pour contrôler l’état initial d’un tel système.

Une étude de l’impact du bruit sur les observations nous permet de conclure que l’espace \mathcal{O} , dans lequel s’effectue la comparaison entre les images observées et les images synthétiques est important. Cette étude montre qu’il est primordial de bien connaître et utiliser les statistiques du bruit corrompant les données quand on souhaite pouvoir exploiter au maximum les informations qu’elles contiennent.

Le fait de travailler dans un espace multi-échelle permet de prendre en compte une partie des corrélations présentes au sein du bruit avec une matrice diagonale. L’intégration de cette information au problème initial permet d’obtenir un problème plus robuste au bruit sans pour autant modifier le coût numérique global.

Ce travail a été financé partiellement par l'Agence Nationale de la Recherche Française à travers le programme COSINUS (projet Geo-FLUIDS n° ANR-09-COSI-005).

Références

- Le Dimet, F.-X. et O. Talagrand (1986). Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus*, 97 – 110.
- Papadakis, N. et E. Mémin (2007). Variational optimal control technique for the tracking of deformable objects. In *International Conference on Computer Vision, ICCV'07*.
- Papadakis, N. et E. Mémin (2008). Variational assimilation of fluid motion from image sequence. *SIAM Journal on Imaging Sciences* 1, 343–363.
- Souopgui, I. (2010). *Assimilation d'images pour les fluides géophysiques*. Ph. D. thesis, Université de Grenoble.
- Titau, O., A. Vidard, I. Souopgui, et F.-X. Le Dimet (2010). Assimilation of image sequences in numerical models. *Tellus Series A : Dynamic Meteorology and Oceanography* 62(1), 30–47.
- Vannucci, M. et F. Corradi (1999). Covariance structure of wavelet coefficients : theory and models in a bayesian perspective. *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 61(4), 971–986.

Summary

Satellite images can provide a lot of information on the earth system evolution. Although those sequences are used more frequently, the importance of spatial error correlation are rarely taken into account in practice. This results in discarding a huge part of the information content of satellite image sequences. In this paper, we investigate a method based on the wavelet transform to represent (at an affordable cost) some of the observation error correlation in a data assimilation context. We address the topic of monitoring the initial state of a system through the variational assimilation of images corrupted by a spatially correlated noise. The feasibility and the reliability of the approach is demonstrated in an academic context.

Adjoint Automatic Differentiation for Data Assimilation

Laurent Hascoët*

*INRIA Sophia-Antipolis, 2004 route des lucioles, 06902 Sophia-Antipolis,
laurent.hascoet@inria.fr,
<http://www-sop.inria.fr/tropics>

Résumé. Nous présentons la Différentiation Automatique (DA), une technique pour obtenir les dérivées d'une fonction fournie sous la forme d'un programme. Nous présentons les principes de la DA, pourquoi elle est possible et pourquoi elle est efficace. Le mode adjoint de la DA est spécialement efficace pour le calcul des gradients tels que ceux qu'utilise l'Assimilation de Données. Nous décrivons la complexité inhérente au mode adjoint, et les outils de DA qui le proposent. Nous montrons que l'utilisation efficace de la DA nécessite un utilisateur averti. L'illustration porte sur deux exemples d'Assimilation de Données en océanographie.

1 Introduction

Data Assimilation is one of the domains of Scientific Computing where derivatives are central : adjusting the data to minimize discrepancy between computed results and measurements literally calls for gradients. Gradients are also central in Inverse Problems and in Optimization. Whereas it is perfectly possible to develop the program that computes these gradients, the mere size and complexity of the system and of its numerical model make automation most welcome. Moreover, numerical models often change to reflect new physics taken into account, and a manually developed gradient would constantly lag behind and never catch up.

Automatic Differentiation (AD) is a set of automated techniques and tools to compute the derivatives of a function F when this function is provided as a computer program P . AD technology can be considered quite mature now, in the sense that many industrial users apply it routinely to their codes. It is also a research domain whose frontiers we are trying to expand. AD can also be read as Algorithmic Differentiation, a slightly less popular but maybe more accurate name for a technique that still requires quite a bit of user insight and intervention. In fact AD is still not, and will probably never be, a push-button solution like a compiler. At least, efficient AD requires a user who understands AD basics and limits, and can drive the AD tool using directives, command-line options or even hand-written selected pieces. Therefore a presentation of AD, like this one, must also describe its basics and its formalization. Further reading should probably start with [Griewank et Walther (2008)], and when AD tool design and architecture is concerned with [Naumann (2012)]. A complement to this paper can also be found in the lecture notes of the Les Houches summer school on Data Assimilation (june 2012), to appear through Oxford University Press.

In section 2 we explain why it is possible to differentiate algorithms, then describe the central tangent mode and adjoint mode of AD, with some emphasis on the latter which is the choice approach for Data Assimilation as discussed in section 3. In section 4 we give a short survey of existing AD tools. Section 5 shows where exactly AD comes into play in Data Assimilation, with two examples showing its performance. In section 6, we conclude by listing a few of the current open problems for research on AD.

2 Basics of Automatic Differentiation

AD is all about derivatives, therefore AD operates on programs implementing functions that are at least once piecewise differentiable. In most cases, these functions exist mathematically as a set of equations, prior to the algorithm that approximates and solves them. However this is not always the case and AD does not require these equations. Instead, AD considers the algorithm as the primary description of the function to differentiate, and builds a new algorithm that computes the derivatives. Obviously, AD makes sense only when the input and output of this algorithm consist mostly of floating-point numbers rather than integers or character strings.

Note that when the problem solved by the program is also expressed as a set of equations, there are basically two ways of obtaining its derivatives, as show on figure 1. One may differentiate the given equations by hand or with a symbolic assistant, and then discretize (i.e. implement) the differentiated equations. Alternatively, one may discretize the original equations, and then use AD to differentiate the discretized algorithm. This second approach spares the effort of implementing the differentiated equations. More surprisingly, it is also seen as more accurate than the first approach, as it maintains consistency between the discretization approximations of the function and of its derivative.

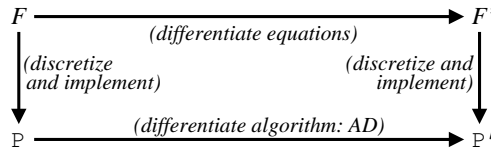


FIG. 1 – Differentiate-then-Discretize vs. Discretize-then-Differentiate

Given a function $F : X \in \mathbb{R}^n \mapsto Y \in \mathbb{R}^m$ in the form of a program piece (i.e. algorithm) \mathbb{P} , AD builds a new program piece \mathbb{P}' that computes derivatives of F by computing the analytical derivative of each elementary mathematical operation in \mathbb{P} . The fundamental justification is that any run-time trace of the algorithm \mathbb{P}

$$\{I_1; I_2; \dots I_p; \}$$

computes the composition of elementary mathematical functions, one per instruction I_k ,

$$f_p \circ f_{p-1} \circ \dots \circ f_1 ,$$

which we can identify to F . This is of course assuming that \mathbb{P} is a correct implementation of F , i.e. the discretization and approximation employed in \mathbb{P} are sufficiently accurate and do not introduce non-differentiability.

Please note the distinction between mathematical variables ($X, Y \dots$) and program variables of \mathbb{P} : as imperative programs classically overwrite their variables to save memory space, one program variable successively represents or holds several mathematical variables. For full generality, we call \mathbb{V} the collection of all the program variables of \mathbb{P} and consider that each instruction I_k (partly) overwrites \mathbb{V} . Thus this run-time trace of \mathbb{P} is indeed the program

original program \mathbb{P}	
	<i>Initialize \mathbb{V} with X</i>
(I_1)	$\mathbb{V} := f_1(\mathbb{V})$
	\dots
(I_k)	$\mathbb{V} := f_k(\mathbb{V})$
	\dots
(I_p)	$\mathbb{V} := f_p(\mathbb{V})$
	<i>Retrieve Y from \mathbb{V}</i>

in which \mathbb{V} holds one particular set (named X_k) of mathematical variables between each instructions I_k and I_{k+1} , and each instruction $(I_k) \mathbb{V} := f_k(\mathbb{V})$ actually takes from \mathbb{V} the mathematical variables X_{k-1} , applies $f_k(X_{k-1})$, and overwrites \mathbb{V} with the resulting X_k .

Now that we have identified F with a composition of functions, the chain rule of calculus gives the first-order full derivative, i.e. the Jacobian :

$$F'(X) = f'_p(X_{p-1}) \times f'_{p-1}(X_{p-2}) \times \dots \times f'_1(X_0) ,$$

so that we can adapt algorithm \mathbb{P} to compute $F'(X)$ in addition to $F(X)$, for instance by inserting before every instruction I_k a new piece of code that computes $J_k = f'_k(X_{k-1}) \times J_{k-1}$. This transformation is local to each instruction I_k . It can be applied to any program \mathbb{P} , even containing control, the extended algorithm \mathbb{P}' just reproducing the control decisions taken by \mathbb{P} . Of course, derivatives are valid only if the control does not change in an open neighborhood around X . Otherwise, the risk is that AD may return a derivative in cases where F is actually non-differentiable. Keeping this caveat in mind the adapted algorithm, given $J_0 = Id$, returns the complete Jacobian $J_p = F'(X)$.

However, the J_k are matrices whose width and height are both of the order of the number of variables in the original \mathbb{P} , and may require too much memory space. Fortunately, it is often sufficient to compute just a few projections of $F'(X)$ such as $F'(X) \times \dot{X}$ for some vector $\dot{X} \in \mathbb{R}^n$ or $\bar{Y} \times F'(X)$ for some $\bar{Y} \in \mathbb{R}^m$. This is in particular true if $F'(X)$ is sparse, in which case a few selected \dot{X} or \bar{Y} can produce all the components of $F'(X)$. This motivates the so-called tangent and adjoint modes of AD :

- **Tangent mode** : evaluate $\dot{Y} = F'(X) \times \dot{X}$, the directional derivative of F along direction \dot{X} . It expands as

$$\dot{Y} = f'_p(X_{p-1}) \times f'_{p-1}(X_{p-2}) \times \dots \times f'_1(X_0) \times \dot{X} . \quad (1)$$

Since \dot{X} is a vector, this formula is most efficiently evaluated from right to left i.e., using mathematical variables :

Adjoint Automatic Differentiation for Data Assimilation

$$\begin{aligned}
 X_0 &= X \\
 \dot{X}_0 &= \dot{X} \\
 X_1 &= f_1(X_0) \\
 \dot{X}_1 &= f'_1(X_0) \times \dot{X}_0 \\
 &\dots \\
 X_k &= f_k(X_{k-1}) \\
 \dot{X}_k &= f'_k(X_{k-1}) \times \dot{X}_{k-1} \\
 &\dots \\
 X_p &= f_p(X_{p-1}) \\
 \dot{X}_p &= f'_p(X_{p-1}) \times \dot{X}_{p-1} \\
 Y &= X_p \\
 \dot{Y} &= \dot{X}_p
 \end{aligned}$$

An algorithm \dot{P} for this evaluation is relatively easy to construct, as the derivative instructions follow the order of the original instructions. Keeping the original program variables \mathbb{V} to hold the successive X_k , and introducing a set of new program variables $\dot{\mathbb{V}}$ of the same size as \mathbb{V} to hold the successive \dot{X}_k , \dot{P} writes :

tangent differentiated program \dot{P}	
	<i>Initialize \mathbb{V} with X and $\dot{\mathbb{V}}$ with \dot{X}</i>
(\dot{I}_1)	$\dot{\mathbb{V}} := f'_1(\mathbb{V}) \times \dot{\mathbb{V}}$
(I_1)	$\mathbb{V} := f_1(\mathbb{V})$
	\dots
(\dot{I}_k)	$\dot{\mathbb{V}} := f'_k(\mathbb{V}) \times \dot{\mathbb{V}}$
(I_k)	$\mathbb{V} := f_k(\mathbb{V})$
	\dots
(\dot{I}_p)	$\dot{\mathbb{V}} := f'_p(\mathbb{V}) \times \dot{\mathbb{V}}$
(I_p)	$\mathbb{V} := f_p(\mathbb{V})$
	<i>Retrieve Y from \mathbb{V} and \dot{Y} from $\dot{\mathbb{V}}$</i>

Notice that each derivative statement \dot{I}_k now precedes I_k , because I_k overwrites \mathbb{V} .

- **Adjoint mode** : evaluate $\bar{X} = \bar{Y} \times F'(X)$, the gradient of the scalar function $\bar{Y} \times F(X)$ derived from F and weights \bar{Y} . It expands as

$$\bar{X} = \bar{Y} \times f'_p(X_{p-1}) \times f'_{p-1}(X_{p-2}) \times \dots \times f'_1(X_0) . \quad (2)$$

Since \bar{Y} is a (row) vector, this formula is most efficiently evaluated from left to right i.e., with mathematical variables :

$$\begin{aligned}
X_0 &= X \\
X_1 &= f_1(X_0) \\
&\dots \\
X_k &= f_k(X_{k-1}) \\
&\dots \\
X_p &= f_p(X_{p-1}) \\
Y &= X_p \\
\bar{X}_p &= \bar{Y} \\
\bar{X}_{p-1} &= \bar{X}_p \times f'_p(X_{p-1}) \\
&\dots \\
\bar{X}_{k-1} &= \bar{X}_k \times f'_k(X_{k-1}) \\
&\dots \\
\bar{X}_0 &= \bar{X}_1 \times f'_1(X_0) \\
\bar{X} &= \bar{X}_0
\end{aligned}$$

However, these formula do not immediately define an algorithm, as the derivative instructions will follow the *inverse* order of the original instructions. Similarly to the tangent mode, we want the adjoint program to use only the original program's variables \mathbb{V} plus a corresponding set of new program variables $\bar{\mathbb{V}}$, of the same size as \mathbb{V} , to hold the successive \bar{X}_k . In that case, we see that e.g. X_{k-1} contained in \mathbb{V} will be overwritten by X_k and thus lost, before it is needed to evaluate $\bar{X}_k \times f'_k(X_{k-1})$. Working around this is called *data-flow reversal*, done through a combination of storage or recomputation of the intermediate X_k . For instance, we show here a “*Store-All*” algorithm that pushes into a stack the variables $\mathbf{out}(I_k)$ overwritten by each (I_k) , and restores them before the corresponding (\bar{I}_k) that executes $\bar{v} := \bar{v} \times f'_k(v)$.

adjoint differentiated program $\bar{\mathbb{P}}$		(Store-All)
	<i>Initialize \mathbb{V} with X and $\bar{\mathbb{V}}$ with \bar{Y}</i>	
	push ($\mathbf{out}(I_1)$)	
(I_1)	$v := f_1(v)$	
	...	
	push ($\mathbf{out}(I_k)$)	
(I_k)	$v := f_k(v)$	
	...	
	push ($\mathbf{out}(I_{p-1})$)	
(I_{p-1})	$v := f_{p-1}(v)$	
(\bar{I}_p)	$\bar{v} := \bar{v} \times f'_p(v)$	
	pop ($\mathbf{out}(I_{p-1})$)	
	...	
	pop ($\mathbf{out}(I_k)$)	
(\bar{I}_k)	$\bar{v} := \bar{v} \times f'_k(v)$	
	...	
	pop ($\mathbf{out}(I_1)$)	
(\bar{I}_1)	$\bar{v} := \bar{v} \times f'_1(v)$	
	<i>Retrieve \bar{X} from \bar{v}</i>	

Notice that the adjoint program exhibits two successive phases : the *forward sweep* is indeed the original \mathbb{P} plus storage of intermediate values ; the *backward sweep* computes the gradients backwards, restoring the intermediate values as it goes.

In adjoint AD the strategy for data-flow reversal is crucial. On *Store-All* approach, we see that the stack grows linearly with the length of \mathbb{P} , which is unbearable for large applications. This is sketched on the left of figure 2, where the arrow on top represents the *forward sweep* with the `push` operations shown as small black dots, whereas the arrow below represents the *backward sweep* with the `pop` operations shown as small white dots. The fundamental tactique is known as *checkpointing*. Checkpointing applied to some part C of \mathbb{P} means doing no `push` during the first sweep on C , but storing instead just enough values to be able to run C again (large black dots). When derivative computation, which goes reverse, reaches the end of C , then C is run again this time with the `push`'es, and then the derivative computation may resume. The gain is that the peak stack size is reduced, roughly by half on figure 2 (middle). The cost is the time to run C again and the relatively small memory or *snapshot* used to run C again. Finally, checkpoints may be nested as sketched on the right of figure 2. Appropriately placed checkpoints yield a peak stack size and an extra recomputation factor that both grow only like the logarithm of the run-time of \mathbb{P} [Griewank (1992)].

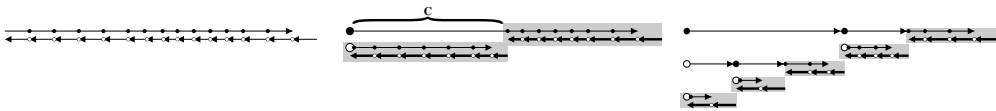


FIG. 2 – Single and nested Checkpointing in the Store-All context

3 Which AD for Data Assimilation ?

Focusing on Data Assimilation allows us to make some choice on the differentiation options. First and foremost Data Assimilation uses derivatives, in fact gradients, to minimize a quadratic cost function that evaluates the squared discrepancy between the model output and the real measurements, possibly mitigated with a given error covariance matrix. This minimization is sensitive to the accuracy of the gradients provided. A relatively small error on the gradient may slow down convergence of the minimization loop, and may simply miss the minimum. For this reason, we advise against approximate derivatives from *divided differences* $(F(x+h) - F(x))/h$. AD derivatives on the other hand are analytic and thus computed with machine accuracy.

Now considering AD, let us compare the run-time costs of the tangent and adjoint modes. Each run of the tangent differentiated algorithm $\dot{\mathbb{P}}$ costs only a small multiple of the run-time of the original \mathbb{P} , with a slowdown ratio typically ranging between 1 and 3. Similarly, the run-time of the adjoint differentiated algorithm $\bar{\mathbb{P}}$ is only a small multiple of the run-time of \mathbb{P} , with a similar ratio, save for the cost of *data-flow reversal*. As the available memory is bounded, the necessary checkpointing implies duplicate executions of selected parts of the code, and the actual slowdown ratio of $\bar{\mathbb{P}}$ compared to \mathbb{P} typically ranges between 5 and 10. Figure 3 helps

us evaluate the costs of computing the complete Jacobian $F'(X)$, using no sparsity property, by employing either $\dot{\mathbb{P}}$ or $\bar{\mathbb{P}}$:

- $\dot{\mathbb{P}}$ gives us $F'(X)$ column by column when \dot{X} ranges over the Cartesian basis of the input space \mathbb{R}^n . The total run-time is typically $3 \times n \times \text{runtime}(\mathbb{P})$.
- $\bar{\mathbb{P}}$ gives $F'(X)$ row by row when \bar{Y} ranges over the Cartesian basis of the output space \mathbb{R}^m . The total run time is typically $10 \times m \times \text{runtime}(\mathbb{P})$.

The choice between tangent and adjoint AD thus depends on the relative dimensions n and m of the input and output spaces of F . When n is much larger than m , the adjoint mode is recommended. In particular in Data Assimilation (also in optimization and inverse problems) m is typically 1 for the scalar discrepancy, whereas n is large as the unknowns to assimilate are functions, shapes, or other complex objects. In that case, no matter the extra cost for data-flow reversal, the adjoint mode of AD is the only reasonable option.

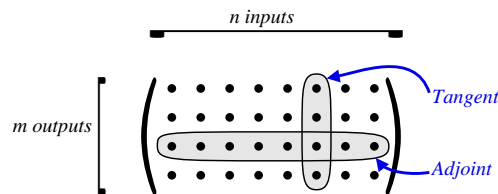


FIG. 3 – Elements of the Jacobian computable by tangent AD and adjoint AD

So much for first-order derivatives. However, a good Data Assimilation often involves an evaluation of the sensitivity of the assimilated data with respect to errors on measurements. Conversely, one may want to know for which measurements an increased accuracy would bring the best accuracy gain on the assimilated data. All these questions of *Uncertainty Quantification* have classical answers, such as the moments method, that involve higher-order derivatives. AD is obviously a choice tool to get these derivatives. For the AD tools that operate by creating a new source program for $\dot{\mathbb{P}}$ and $\bar{\mathbb{P}}$, repeated application of the tool can produce $\dot{\dot{\mathbb{P}}}$, $\bar{\bar{\mathbb{P}}}$, $\dot{\bar{\mathbb{P}}}$... to compute second derivatives. The choice of the best option depends on the structure of \mathbb{P} and on the kind of derivatives needed for the specific uncertainty analysis.

4 AD tools

Here is a short survey of available AD tools and of how one might classify them. The best source is the www.autodiff.org site, which is the portal managed by our colleagues in Aachen and Argonne in the name of the AD community. It turns out that plenty of experiments were made that adapt the concepts of AD to particular environments. There was for instance a clever introduction of tangent AD into Microsoft's Excel. There are also tools that target higher-level languages, such as MATLAB. We know of ADiMat, MAD, and INTLAB. However in the present context, we will focus on the tools that we feel can be applied to Data Assimilation and inverse problems of industrial size. Essentially, there are two principal ways for a tool to provide the algorithms $\dot{\mathbb{P}}$ and $\bar{\mathbb{P}}$ namely, *Operator Overloading* and *Source Transformation*.

Operator Overloading : if the language of \mathbb{P} permits, one can replace the types of the floating-point variables with a new type that contains additional derivative information, and overload the arithmetic operations for this new type so as to propagate this derivative information along. Schematically, the AD tool boils down to a library that defines the overloaded type and arithmetic operations. This approach is both elegant and powerful, and such tools are relatively easy implemented. The overloaded library can be quickly redefined to compute higher-order derivatives, Taylor expansions, intervals. . . By nature, evaluation of the overloaded operations will follow the original order of \mathbb{P} . This is fine for the tangent mode, but requires some acrobacy for the adjoint mode, with a cost in performance, although some tools get very sophisticated and efficient, and represent a fair bit of development work. Just a few examples :

- **Adol-C** [Walther et Griewank (2012)] may be the most sophisticated. It applies to C++ source, although something did exist for Fortran90. It produces many different sorts of derivatives and has been tested extensively on many large codes using some form of checkpointing.
- **dco** [Lotz et al. (2011)] is more recent and applies to C++. Initially designed as a teaching tool with some simplifying assumptions, it is now applicable to industrial codes with good performance.
- **Rapsodia** [Charpentier et Utke (2009)] is a tool that generates Overloading AD libraries from a higher-level description.
- **MAD** [Forth (2006)] applies Overloading AD to MATLAB codes, using a deep understanding of MATLAB internal structures to achieve efficiency.
- **The complex-step** [Martins et al. (2003)] is a clever use of properties of Complex numbers : redeclaring every `real` variable as a `complex`, and using the imaginary part of each variable v to hold its tangent derivative \dot{v} , scaled to be small enough so that its square falls below machine precision when added to v^2 , one obtains a good enough approximation of tangent AD.

Source Transformation : Other tools can choose to explicitly build a new source code that computes the derivatives. This means to parse the original \mathbb{P} and to build an internal representation, and from it to build the differentiated $\dot{\mathbb{P}}$ or $\bar{\mathbb{P}}$. This approach allows the tool to apply some global analysis on \mathbb{P} , for instance data-flow, to produce a more efficient differentiated code. This is very similar to a compiler, except that it produces source code. This approach is more development-intensive than Overloading AD, which is one reason why Overloading AD tools appeared earlier and are more numerous. It also explains why Source Transformation AD tools are often more fragile and need more effort to follow the continuous evolution of programming constructs and styles. On the other hand, the possibility of global analysis makes Source Transformation the choice approach for the adjoint mode of AD, which requires control-flow and data-flow reversal and where global analysis is essential to produce efficient code. The following tools differ in particular in the language that they recognize and differentiate, and in the AD modes that they provide. They also exhibit some differences in AD strategies mostly about the adjoint mode. Here are a few examples :

- **Tapenade** [Hascoët et Pascual (2013)] provides tangent and adjoint differentiation of Fortran (77 and 95) and C. Adjoint mode uses the Store-All approach to restore intermediate values. This Store-All approach is selectively replaced by recomputation in simple appropriate situations. Checkpointing is applied by default at the level of procedure calls, but can be triggered or deactivated at other places through user directives.

- **Adifor** [Carle et Fagan (2000)] differentiates Fortran77 codes in tangent mode. Adifor once was extended towards the adjoint mode (Adjfor), but we believe this know-how has now been re-injected into the OpenAD framework, described below.
- **OpenAD/F** [Utke et al. (2008)] differentiates Fortran codes in tangent and adjoint modes. The general framework of OpenAD claims (like Tapenade) that only front-end and back-end should depend on the particular language, whereas the analysis and differentiation part should work on a language-independent program representation. In that sense OpenAD is in fact the fusion of former tools Adifor, Adjfor, and Adic, integrated into the OpenAD framework. Its strategy to restore intermediate values in adjoint AD is extremely close to Tapenade’s.
- **TAMC** [Giering (1997)], through its commercial offsprings TAF and TAC++ differentiates Fortran and C files. TAF also differentiates Fortran95 files, under certain restrictions. TAF is commercialized by the FastOpt company in Hamburg, Germany. Differentiation is provided in tangent and adjoint mode. Intermediate values in adjoint AD are restored in an alternative way known as *Recompute-All*. Checkpointing and an algorithm to avoid useless recomputations (ERA) are used to avoid explosion of run-time. TAF also provides a mode that efficiently computes the sparsity pattern of Jacobian matrices, using bit-sets.

There are also AD tools that directly interface to an existing compiler. In fact, these are modifications to the compiler such that the compiler performs AD at compile time. This new AD functionality is triggered by new constructs or new directives added into the application language. In a sense, this amounts to modifying the application language so that it has a new “differentiation” operator. For instance the NAGWare Fortran95 compiler embeds AD facilities [Naumann et Riehme (2005)], that are triggered by user directives in the Fortran source. To our knowledge, its adjoint mode uses a strategy equivalent to Overloading AD.

5 Application to Data Assimilation

In Data Assimilation, we have a physical model that represents the way some unknown values determine some measurable values. When this model is complex, its inverse problem is nontrivial. From this physical model we get a mathematical model, which is in general a set of partial differential equations, and this mathematical model has been discretized and solved, in other words implemented as a program \mathbb{P} .

Using the formalism of *optimal control theory* [Le Dimet et Talagrand (1986)], we are studying the state W of some system. W is defined for every point in space and also – if time is involved – for every instant in an observation period $[0, T]$. The mathematical model relates W to a number of external parameters, which are the collection of initial conditions, boundary conditions, model parameters, etc, i.e. all the values that determine the state. Some of these parameters, that we call γ , are the unknown of our data assimilation problem, whereas $W = W(\gamma)$ is a function of γ defined in general implicitly by a set of partial differential equations that we write :

$$\Psi(\gamma, W) = 0 \quad . \quad (3)$$

In optimal control theory, we would call γ our control variable. From $W(\gamma)$, we can extract the measurable values, which of course do not exactly match the values actually measured W_{obs} .

Adjoint Automatic Differentiation for Data Assimilation

Therefore we start an optimization cycle to modify γ , until the resulting measurable values match best : we define a *cost function* that measures the discrepancy on the measurable values of $W(\gamma)$. In practice, not all values in $W(\gamma)$ can be measured in W_{obs} , but nevertheless we can define this cost function $j(\gamma) = J(W(\gamma))$ as the sum at each instant of some squared discrepancy, possibly involving an error covariance matrix C , between the computed and measured values $(W(\gamma) - W_{obs})^* C (W(\gamma) - W_{obs})$.

The inverse problem is to find the value of γ that minimizes $j(\gamma)$, i.e. such that $j'(\gamma) = 0$. If we use a gradient descent algorithm to find γ , we need $j'(\gamma)$ for each γ . To this end, the mathematical approach first applies the chain rule to the definition of $j(\gamma)$, yielding :

$$j'(\gamma) = \frac{\partial J(W(\gamma))}{\partial \gamma} = \frac{\partial J}{\partial W} \frac{\partial W}{\partial \gamma} . \quad (4)$$

The derivative of W with respect to γ comes from the state implicit equation (3), which we differentiate with respect to γ to get :

$$\frac{\partial \Psi}{\partial \gamma} + \frac{\partial \Psi}{\partial W} \frac{\partial W}{\partial \gamma} = 0 . \quad (5)$$

Assuming this can be solved for $\frac{\partial W}{\partial \gamma}$, we then substitute it into equation (4) to get :

$$j'(\gamma) = - \frac{\partial J}{\partial W} \frac{\partial \Psi}{\partial W}^{-1} \frac{\partial \Psi}{\partial \gamma} . \quad (6)$$

Now is the time to take complexity into account. Equation (6) involves one system resolution and then one product. Nowadays both Ψ and W are discretized with millions of dimensions. $\frac{\partial \Psi}{\partial W}$ is definitely too large to be computed explicitly, and therefore its inverse cannot be computed nor stored either. We notice that $\frac{\partial \Psi}{\partial \gamma}$ has many columns, whereas $\frac{\partial J}{\partial W}$ has only one row. Therefore the most efficient way to compute the gradient $j'(\gamma)$ is the *adjoint method* : first compute $\frac{\partial J}{\partial W}$, then run an iterative resolution for

$$\frac{\partial J}{\partial W} \frac{\partial \Psi}{\partial W}^{-1} \quad (7)$$

and then multiply the result (called the *adjoint state* Π) by $\frac{\partial \Psi}{\partial \gamma}$.

We might now set off for a long implementation session to write the program that discretizes and solves equation (7) but *hold on, wait a minute!*.. Instead of that, recall that we already have a resolution program, i.e. a procedure P_Ψ which, given γ , returns W_γ , and a procedure P_j which, given a W , evaluates the cost function, i.e. the discrepancy between W and the observed W_{obs} . Adjoint AD of the program that computes

$$j = P_j(P_\Psi(\gamma))$$

directly gives the gradient of j , i.e. the desired $j'(\gamma)$. This is indeed very close to the mathematical resolution with the adjoint state. In both cases, $\frac{\partial J}{\partial W}$ is computed first, thus guaranteeing that no large square matrix is stored. Things differ a little for the following stage, because a program is by essence explicit and therefore the resolution for Π and multiplication with $\frac{\partial \Psi}{\partial \gamma}$

are done jointly. Apart from that, adjoint AD can really be considered as the discrete equivalent (on programs) of the above adjoint method.

As a first application, we considered the oceanography code OPA 9.0 [Madec et al. (1998)] on a simple configuration known as GYRE. It simulates the behavior of a rectangular basin of water put on the tropics between latitudes 15° and 30° , with the wind blowing to the East. Our control variables γ are the temperature field in the complete domain, and our cost function $j(\gamma)$ is the discrepancy with respect to measurements of the heat flux across some boundary at the northern angle 20 days later. Figure 4 shows one gradient $j'(\gamma)$ computed by adjoint AD. This

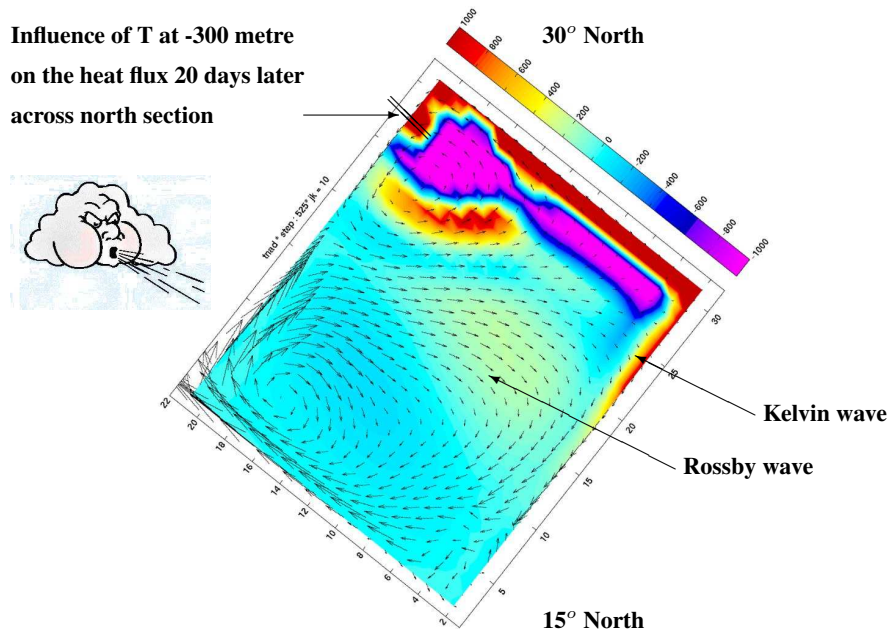


FIG. 4 – Oceanography gradient by adjoint AD on OPA Gyre

system is discretized with $32 \times 22 \times 31$ nodes and 4320 time steps. The original simulation takes 26 seconds, and the differentiated program computes the gradient above in 205 seconds, which is only 7.9 times as long. Of course checkpointing is absolutely necessary to reverse this long simulation, yielding several recomputations of the same program steps, but nevertheless the factor 7.9 is much better than what tangent AD would require. Checkpointing and storage of the trajectory use a stack that reaches a peak size of 481 Mbytes.

In a second, larger application we considered the NEMO configuration of OPA (2° grid cells, one year simulation) for the north Atlantic. Figure 5 shows the gradient of the discrepancy to measurements of the heat flux across the 29^{th} parallel, with respect to the temperature field one year before. With this discretization, there are 9100 control parameters in γ . The adjoint code that computes the gradient with respect to these 9100 parameters takes 6.5 times as long as the original code. The size of the stack that stores the trajectory and the snapshots for checkpointing reaches a peak of 1591 Mbytes.

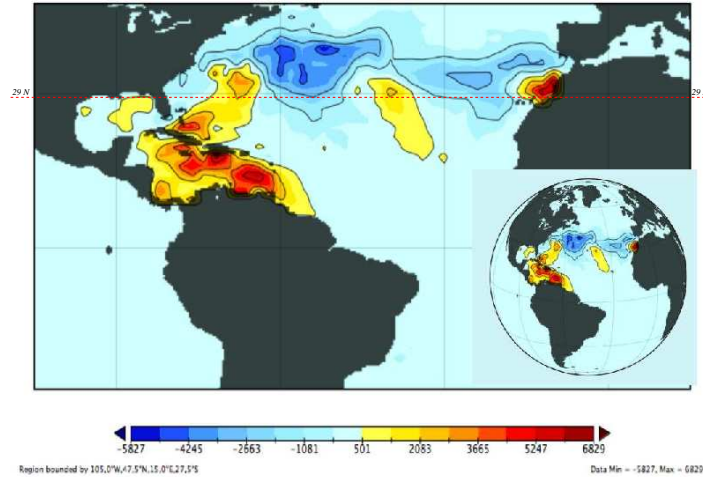


FIG. 5 – Oceanography gradient by adjoint AD on OPA Nemo

6 Conclusion and Frontiers of Automatic Differentiation

We briefly described the mathematical justification of Automatic Differentiation, focusing on the tangent and adjoint first-order derivatives. We explained why the adjoint mode is the choice approach for Data Assimilation and we showed a couple of applications from Oceanography. However, the adjoint mode of AD is also inherently complex and we showed how existing AD tools work to provide it. We encourage researchers to use AD for Data-Assimilation, selecting one or several in the long list of existing AD tools. These tools make steady progress and are developed by teams who are eager for collaboration. We will conclude with a word on the ongoing research on AD, although it might mitigate the enthusiasm that you certainly now feel about AD. AD is facing a few research problems. Some of them are definitely fundamental, others are mostly a question of time for needed research and development.

On the fundamental side is the question of non-differentiability of control. Programs contain control in the form of conditionals or in the form of loops, that make them differentiable only piecewise. Depending on the programming style, even differentiable functions can be implemented as a non-differentiable program, for instance when solving with the bisection method. There exist only partial answers to this problem : some differentiated codes emit a message at run-time when some test is “close” to switching. Others return an additional indicator of the interval around X and along direction \dot{X} in which no control switches. Some researchers think of building a formal certificate that a differentiated code doesn’t switch for a certain domain of X . Finally, some want to develop new derivative-based optimization schemes that would accept a class of non-smooth functions. In most cases, real-life codes are cluttered with control and testing the validity of their AD derivatives is still an absolute necessity.

A special case of this control problem lies with iterative loops, for instance fixed-point iterations. In most cases, one can show that the convergence rate of the differentiated iteration

is similar to the original. However, one can build contrived examples where the derivative has not converged when the original value has, because the starting point was chosen too close to the fixed point. Automated answers to this question still seem very crude and would reject many programs as non-differentiable. A better answer is to interact with the end-user, who may use directives to point the AD tool at the iterative loops and to trigger specialized strategies. Moreover, this is beneficial in terms of performance : a special strategy for the adjoint of a Newton solver will only need to adjoin the last step of the Newton process, therefore saving much derivative computation. Likewise, a special strategy to differentiate an iterative linear solver will simply call the same solver on different inputs, not differentiating the tricks of the trade inside the solver.

Still rather fundamental is the problem of data-flow reversal of very long simulations, but we believe that this is correctly answered now by a collection of user directives to fine-tune checkpointing. Of course simulations can always get longer so that data-flow reversal runs short of memory, but the same may occur to hand-written adjoints so that the problem is no longer specific to AD.

On the less fundamental side are the questions related to “modern” programming styles and constructs. For historical reasons, many AD tools target Fortran and didn’t care very much about constructs such as dynamic memory allocation, polymorphism, or Object-Oriented style. These cause problems with the adjoint mode of AD. Overloading AD tools enjoy a somewhat better situation there, as their strategy isolates them from the exotic constructs found in modern languages. In fact they even *need* to apply to these modern languages, since they need operator overloading. Source transformation AD tools need more time to catch up. For instance, we are getting very close to a general approach to adjoint AD of dynamic memory and pointers management. The interesting bit is that this will probably be a run-time approach akin to what Overloading AD does, because static analysis as performed by Source-Transformation is in general unable to extract the needed information at transformation time.

Parallelism is another sort of a modern construct that AD tools must handle better. The problem is faced by Overloading AD tools as well. For instance for Message-Passing, current research goes towards building a special wrapper around existing libraries (e.g. MPI) that will monitor communication of derivatives at run-time.

Références

- Carle, A. et M. Fagan (2000). ADIFOR 3.0 overview. Technical Report CAAM-TR-00-02, Rice University.
- Charpentier, I. et J. Utke (2009). Rapsodia : User manual. Technical report, Argonne National Laboratory.
- Forth, S. (2006). An efficient overloaded implementation of forward mode automatic differentiation in MATLAB. *ACM Transactions on Mathematical Software* 32(2), 195–222.
- Giering, R. (1997). Tangent linear and adjoint model compiler, users manual. Technical report. <http://www.autodiff.com/tamc>.
- Griewank, A. (1992). Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and Software* 1, 35–54.

- Griewank, A. et A. Walther (2008). *Evaluating Derivatives : Principles and Techniques of Algorithmic Differentiation*. Other Titles in Applied Mathematics, #105. SIAM.
- Hascoët, L. et V. Pascual (2013). The Tapenade Automatic Differentiation tool : Principles, Model, and Specification. *ACM Transactions On Mathematical Software* 39(3).
- Le Dimet, F.-X. et O. Talagrand (1986). Variational algorithms for analysis and assimilation of meteorological observations : theoretical aspects. *Tellus* 38A, 97–110.
- Lotz, J., K. Leppkes, et U. Naumann (2011). dco/c++ - derivative code by overloading in C++. Technical report, Aachener Informatik-Berichte (AIB).
- Madec, G., P. Delecluse, M. Imbard, et C. Levy (1998). OPA8.1 ocean general circulation model reference manual. Technical report, Pole de Modelisation, IPSL.
- Martins, J., P. Sturdza, et J. Alonso (2003). The Complex-Step derivative approximation. *ACM Transactions on Mathematical Software* 29(3), 245–262.
- Naumann, U. (2012). *The Art of Differentiating Computer Programs : An Introduction to Algorithmic Differentiation*. Software, Environments, and Tools, #24. SIAM.
- Naumann, U. et J. Riehme (2005). Computing adjoints with the NAGWare Fortran 95 compiler. In Bücker, Corliss, Hovland, Naumann, Norris (Ed.), *Automatic Differentiation : Applications, Theory, and Implementations*, Volume 50. Springer LNCSE.
- Utke, J., U. Naumann, M. Fagan, N. Tallent, M. Strout, P. Heimbach, C. Hill, et C. Wunsch (2008). OpenAD/F : A modular, open-source tool for Automatic Differentiation of Fortran codes. *ACM Transactions on Mathematical Software* 34(4), 18 :1–18 :36.
- Walther, A. et A. Griewank (2012). Getting started with ADOL-C. In Naumann and Schenk (Ed.), *Combinatorial Scientific Computing*, Chapter 7, pp. 181–202. Chapman-Hall CRC Computational Science.

Summary

We present Automatic Differentiation (AD), a technique to obtain derivatives of functions provided as programs. We present the principles that justify why AD is possible and explain its performance. The adjoint mode of AD is the choice approach to obtain gradients, like the gradients needed for data assimilation. We show the specific difficulties of the adjoint mode, and list a few AD tools that handle these problems well. We show why AD needs an enlightened user to achieve optimal efficiency. Two examples from Oceanography illustrate the use and performance of AD for Data Assimilation.

Prise en compte de structures spatiales pour l'assimilation variationnelle de données de télédétection. Exemple sur un modèle simple de croissance de végétation.

Claire Lauvernet *, François-Xavier Le Dimet**
Frédéric Baret***

*Irstea, UR MALY, centre de Lyon-Villeurbanne
5 rue de la Doua-CS 70077
F-69626 Villeurbanne cedex, France
claire.lauvernet@irstea.fr
<http://www.irstea.fr/pollutions-agricoles-diffuses>

**Université de Grenoble, France
Francois-Xavier.Le-Dimet@imag.fr
<http://www-ljk.imag.fr>

***INRA-EMMAH UMR 1114, France
baret@avignon.inra.fr

Résumé. En assimilation de données, une série temporelle de données-image devrait être traitée explicitement pour en extraire toute l'information. Cette étude propose d'adapter une méthode d'assimilation variationnelle d'images de LAI (Leaf Area Index) dans un modèle de végétation, afin d'intégrer l'information liée à l'aspect spatial des données. Pour cela, on considère que les paramètres sont contrôlés spatialement à certains niveaux: variété, parcelle, pixel, ou stables temporellement sur l'ensemble de l'image. Ces contraintes réduisent la taille du problème inverse, en transformant le schéma d'assimilation habituel à des ensembles de pixels simultanés. La méthode est appliquée sur le modèle semi-mécaniste BONSAÏ et évaluée sur la qualité de prédiction du LAI et d'estimation des paramètres d'entrée par expériences jumelles, ainsi que sur sa sensibilité à la fréquence des observations. Les contraintes spatio-temporelles améliorent la robustesse et les estimations lorsque la quantité d'observations disponibles diminue, par rapport à la méthode classique, où chaque pixel.date est considéré indépendamment des autres.

1 Introduction

Dans les applications de méthodes d'assimilation de données de télédétection dans le domaine agronomique, l'approche considère généralement le problème pour chaque pixel.date indépendamment des autres, bien que l'on assimile des données-image. Le LAI est tout d'abord obtenu typiquement par inversion d'un modèle de transfert radiatif sur chaque pixel. Ceci signifie que, à une date donnée, on considère une pseudo-observation de LAI sur un pixel (ou une

moyenne de pixels proches) que l'on assimile ensuite dans un modèle de croissance de plante. Certaines contraintes sont parfois ajoutées afin de conditionner le comportement d'un pixel par celui des pixels voisins pour assurer une cohérence spatiale, mais pas sur les dépendances spatiales des paramètres à identifier. Le fait d'estimer les paramètres d'un pixel indépendamment des pixels voisins génère au moins 4 critiques :

- on ne prend pas en compte les relations de structures spatiales, alors qu'il n'y a pas forcément indépendance entre les variables décrivant les phénomènes en divers points de l'espace
- on perd de l'information relative au fait d'utiliser des images, et même des séries temporelles d'images, ce qui constitue pourtant une source d'information supplémentaire à exploiter
- le problème inverse est généralement mal posé
- on répète la même action un très grand nombre de fois, induisant probablement un coût de calcul non optimal.

Dans les problèmes d'inversion de modèles de taille conséquente, le coût de calcul devient un véritable obstacle, et le minimiser représente un objectif supplémentaire. Certaines contraintes sont parfois ajoutées afin de conditionner le comportement d'un pixel par celui des pixels voisins pour assurer une cohérence spatiale, mais pas sur les dépendances spatiales des paramètres à identifier. De fait, on peut se poser les questions suivantes :

1. quelles structures spatiales exploiter pour réduire la taille du problème et le rendre inversible ?
2. comment transformer le schéma d'assimilation habituel à un ensemble de pixels ?

Cette étude propose une approche de prise en compte de structures spatiales des paramètres d'entrée d'un modèle de croissance de la végétation lors de la mise en place d'une méthode d'assimilation de données image.

2 Matériel et méthodes

2.1 Le modèle agronomique Bonsaï

Le modèle utilisé dans cette étude est particulièrement simple par rapport à des modèles mécanistes plus complexes de fonctionnement du couvert végétal (Brisson, 1998, 2002). Ce choix est justifié par la volonté de tester l'intérêt de la prise en compte de contraintes spatiales en maîtrisant mieux l'outil, en limitant le nombre de ses paramètres, et en simplifiant la différenciation nécessaire à la mise en place d'une méthode variationnelle.

Le modèle BONSAÏ est un modèle semi-mécaniste, adapté de MODLAI (Baret, 1986). Il simule l'indice foliaire (surface totale d'éléments de végétation verts par unité de surface au sol horizontal) en fonction du temps exprimé en degrés.jours cumulés :

$$\left\{ \begin{array}{l} LAI_t = L_{max} \cdot \left(\frac{1}{(1+e^{-A(ST_t-t_0-Ti)})^C} - e^{B(ST_t-t_0-(\Delta Ts+Ti))} \right) \\ A = \frac{1}{Ti} \cdot \log \left(\left(e^{\frac{B}{C} \cdot (\Delta Ts+Ti)} \right) - 1 \right) \\ \text{si} \quad \left(e^{\frac{B}{C} \cdot (\Delta Ts+Ti)} \right) < 1, \text{ alors } \left(e^{\frac{B}{C} \cdot (\Delta Ts+Ti)} \right) = 2 \end{array} \right. \quad (1)$$

où t représente le temps (degrés.jours cumulés depuis la levée), LAI_t est l'indice foliaire au jour t , ST_t est la somme des degrés.jours cumulés depuis la date de levée jusqu'au jour t , T_i est le seuil de degrés.jours cumulés de croissance maximale, ΔT_s est la somme de degrés.jours entre le point de croissance maximale et la sénescence, B représente la vitesse de sénescence, L_{max} est le LAI maximal, C représente le point d'inflexion dans la partie croissante, t_0 est la date de levée.

2.2 Structures spatiales exploitées

Pour réduire la taille du problème, on fait l'hypothèse forte que les paramètres du modèle dépendent chacun d'un certain niveau spatial. On parlera par exemple de paramètre *variétal* lorsque celui-ci reste fixe pour une même variété, comme par exemple les stades phénologiques ou les propriétés des feuilles. A un degré inférieur, certains paramètres dépendent de la *parcelle*, comme les pratiques culturales, même si ce choix est discutable pour l'irrigation, qui n'est pas toujours homogène sur une parcelle. Dans cette étude, basée sur un site d'expérimentation où les terrains sont plats et l'irrigation très faible, cette hypothèse est raisonnable. D'autres paramètres dépendent du *pixel*, tels que les paramètres du sol. Enfin, pour améliorer la robustesse de la méthode et réduire l'espace de contrôle, on considère certains paramètres comme fixes sur toutes les parcelles et pour toutes les variétés, en particulier les paramètres auxquels le modèle montre peu de sensibilité, et, dans le cas de BONSAÏ, les paramètres n'ayant aucune signification physique et dont on ne peut décider s'ils suivent une loi particulière. On les dira *spécifiques* relativement à l'expérience considérée. On pourrait également inclure un groupe de paramètres fixes sur certaines périodes temporelles puisque l'on assimile une série temporelle d'images. BONSAÏ n'est pas décrit par ce type de paramètres, mais cela a été fait par exemple dans Lauvernet et al. (2008). On résume dans le tableau 1 les différents niveaux de dépendance possibles :

paramètres	caractéristique spatiale	exemple de param.
spécifiques	égaux sur toutes les parcelles	sans signification
variétaux	égaux sur les parcelles d'une même variété	stades phénol.
parcellaires	égaux sur les pixels d'une même parcelle	techn. cult.
pixellaires	dépendent du pixel	propriétés du sol

TAB. 1 – Définition des différents niveaux de dépendance spatiale des paramètres.

Dans l'ensemble de ce travail, l'expression *contraintes spatiales* ne signifie pas que l'on impose des contraintes aux pixels en rapport à leur distance ou connectivité les uns aux autres. Elle doit être interprétée comme des contraintes liées par une modélisation des paramètres caractérisée par leur appartenance à des pixels d'une image, avec des effets fixes parcelles, variétaux et des effets locaux relatifs au sol.

2.3 Assimilation de données variationnelle avec contraintes spatiales

La méthode d'assimilation de données développée dans cette étude est une méthode variationnelle utilisant le modèle adjoint pour minimiser la fonction coût représentant l'écart entre

les observations (ici, de LAI) et les simulations du modèle. De nombreuses références décrivent la théorie de cette méthode qui ne sera donc pas développée ici (Le Dimet et Talagrand, 1986). D'un point de vue pratique, on distingue quatre étapes fondamentales dans le développement de la méthode d'assimilation par l'adjoint : (1) résoudre le modèle direct (2) résoudre le modèle adjoint (3) en déduire le gradient de la fonction coût J en fonction de ce que l'on veut estimer (4) fournir ce gradient à un algorithme de minimisation pour en déduire les variables de contrôle optimales.

En assimilation de données classique (ou pixel par pixel), la fonction coût à minimiser est donnée par :

$$J(\theta_{est}) = \sum_{i=1}^N |LAI(i) - LAI_{obs}(i)|^2 + \frac{\|\theta_{est} - \theta_{ap}\|^2}{\sigma_{ap}^2}$$

avec θ_{est} le vecteur des paramètres estimés par l'algorithme, de dimension 6, N le nombre d'observations assimilées, LAI l'indice foliaire simulé par le modèle, LAI_{obs} l'indice foliaire observé. Le terme d'information *a priori* (IAP) est représenté par l'écart entre les paramètres estimés θ_{est} et les valeurs *a priori* des paramètres θ_{ap} , pondéré par les variances *a priori* associées σ_{ap} .

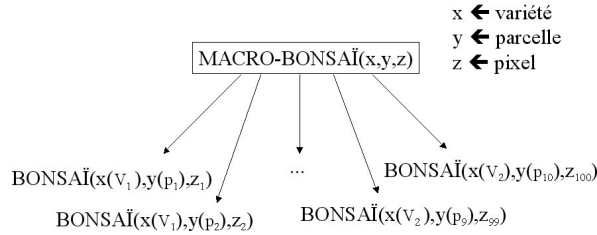


FIG. 1 – Représentation du modèle MACRO-BONSAÏ prenant en compte trois niveaux de dépendance des paramètres de BONSAÏ : la variété, la parcelle, et le pixel.

Une fois la caractérisation des propriétés spatiales des paramètres de BONSAÏ déterminée, on peut créer le modèle MACRO-BONSAÏ (représenté FIG. 1), qui consiste à exécuter BONSAÏ un certain nombre de fois selon le nombre de pixels, parcelles et variétés que l'on évalue, en prenant en compte les contraintes dans leur globalité. Pour cela, on considère le LAI comme fonction de trois vecteurs de paramètres : x dépendant de la variété, y de la parcelle, et z du pixel :

$$LAI = BONSAÏ(x, y, z)$$

puis, on contraint le vecteur de paramètres x à être constant pour tout pixel de même variété, et le vecteur de paramètres y à être constant sur une même parcelle. On pourra également ajouter un autre vecteur de paramètres spécifiques à l'expérience si besoin, c'est-à-dire d'une valeur égale partout. Ce "macro-modèle" sera alors dérivé par rapport à l'ensemble de ses paramètres, afin d'obtenir le modèle adjoint permettant de minimiser la fonction coût adaptée également, pour dépendre de tous les paramètres de MACRO-BONSAÏ que l'on souhaite identifier. On ajoute à la fonction coût un terme Bayésien permettant de prendre en compte l'information a

priori sur chaque type de paramètres et régulariser ainsi le problème inverse :

$$J(x, y, z) = \sum_{i=1}^N \frac{\|LAI(i) - LAI_{obs}(i)\|^2}{\sigma_{LAI}^2} + \sum_{v=1}^{nbVar} \frac{\|x(v) - x^0(v)\|^2}{\sigma_{var}^2(v)} + \sum_{par=1}^{nbPar} \frac{\|y(par) - y^0(par)\|^2}{\sigma_{par}^2(par)} + \sum_{pix=1}^{nbPix} \frac{\|z(pix) - z^0(pix)\|^2}{\sigma_{pix}^2(pix)}$$

où :

- $nbVar, nbPar, nbPix \in \mathbb{N}$ correspondent respectivement aux quantités de variétés, parcelles, et pixels ; $N \in \mathbb{N}$ est le nombre d'observations total (sur tous les pixels) de LAI ;
- $i \in \{1, \dots, N\}$, $v \in \{1, \dots, nbVar\}$, $par \in \{1, \dots, nbPar\}$, $pix \in \{1, \dots, nbPix\}$ des indices entiers naturels ;
- $\sigma_{LAI}^2 \in \mathbb{R}$ est la variance associée à l'incertitude des pseudo-observations de LAI ;
- $\sigma_{var}^2, \sigma_{par}^2, \sigma_{pix}^2$ sont les variances *a priori* associées respectivement aux paramètres variétaux, parcellaires et pixellaires.

Le modèle macro-BONSAÏ devra donc être dérivé par rapport à tous les paramètres de la fonction coût J , c'est-à-dire par rapport à un nombre de $x * nbVar + y * nbPar + z * nbPix$ variables.

3 Évaluation de la méthode par des expériences jumelles

Le modèle BONSAÏ est un modèle simple et permettant de calculer facilement l'adjoint. Toutefois, certains de ses paramètres sont de fait difficilement hiérarchisables par rapport à une dépendance spatiale. Nous avons donc décidé d'exploiter les expériences jumelles pour évaluer l'intérêt de l'assimilation avec contraintes spatiales dans un premier temps, sur deux variétés virtuelles. Ainsi nous forçons par les données l'aspect variabilité spatiale pour voir l'effet sur l'assimilation malgré la simplicité du modèle.

3.1 Création des données virtuelles

Les niveaux de dépendance spatiale des paramètres sont à définir à la fois selon la connaissance des processus et du modèle. Pour BONSAÏ, le tableau 2 donne les niveaux de dépendance des paramètres, basés sur l'expertise, et qui seront donc utilisés pour les expériences jumelles.

Pour générer les observations virtuelles de LAI (tableau 3), on a créé 2 variétés (par les paramètres T_i et ΔT_s) de céréale cultivée se comportant très différemment sur le cycle, la phase de croissance de la variété 2 recoupant en partie la phase de sénescence de la variété 1. On a ensuite créé 10 parcelles par variété, avec pour chaque parcelle, la date de levée t_0 définie de manière aléatoire sur son intervalle de définition. Puis 10 pixels ont été générés par parcelle, avec sur chaque parcelle, les paramètres dépendant du pixel tels que B , C et L_{max} prenant chacun 10 valeurs différentes, choisies aléatoirement sur leurs intervalles de définition.

Les observations de LAI ont donc été créées à partir de ces jeux de paramètres sur les 100 pixels, puis bruitées de 10% selon une loi gaussienne de moyenne 0 et écart-type 1 (FIG. 4). Les indices foliaires sont représentés FIG. 3, sur les 100 pixels (ordonnée), en fonction des

Assimilation d'images sous contraintes spatiales dans un modèle agronomique

Param.	dépendance
t_0	parcelle
T_i	variété
ΔT_s	variété
B	pixel
L_{max}	pixel
C	pixel

TAB. 2 – Attribution d'un niveau de dépendance de chaque paramètre du LAI dans BONSAÏ.

vecteur	nb.param (A)	quantité (B)	dim=AB
x_{parcelle}	1	10	10
$y_{\text{variété}}$	2	2	4
z_{pixel}	3	100	300

TAB. 3 – Quantités des paramètres générés par les expériences jumelles, par vecteur (type de dépendance spatiale). Au total, on a $10+4+300 = 314$ paramètres d'entrée.

jours (abscisse). Les deux variétés se distinguent particulièrement bien entre les 50 premiers pixels et les 50 derniers. Chaque dizaine de pixels représente une parcelle et on constate ainsi que les dates de levée y sont constantes.

Notons que pour une visualisation spatio-temporelle permettant de constater la bonne (ou mauvaise) représentativité des parcelles et variétés, les résultats après assimilation seront toujours comparés aux observations de référence, sans bruit ajouté (FIG. 3).

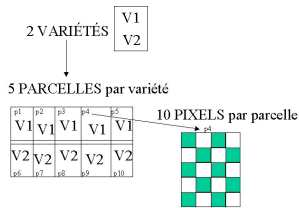


FIG. 2 – Représentation des trois échelles : la variété, la parcelle et le pixel.

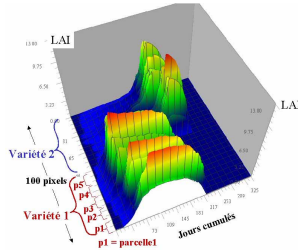


FIG. 3 – Représentation des LAI pour les 2 variétés, sur les 10 parcelles.

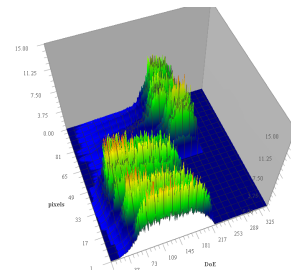


FIG. 4 – Observations de LAI simulées bruitées.

3.2 Différentiation de MACRO-BONSAÏ

Pour résoudre le problème de minimisation, l'adjoint de MACRO-BONSAÏ est calculé avec l'outil de différentiation automatique Tapenade (Hascoët et al., 2001), en fonction des trois vecteurs de paramètres. Ainsi, sans la prise en compte des contraintes, on assimile successivement sur les 100 pixels, ce qui représente l'estimation de 600 paramètres. Par contre, avec les contraintes spatiales, on résout un problème de taille 314 seulement (voir tableau 3). Pour la minimisation de la fonction coût, l'algorithme de minimisation NIQN3 a été utilisé (Gilbert et Lemaréchal, 1989).

3.3 Évaluation des expériences jumelles sur les estimations de LAI

La représentation spatio-temporelle des valeurs de LAI après assimilation a pour seule fonction de vérifier que les niveaux de dépendance parcelles et variétés sont bien conservés. On constate que les LAI obtenus après assimilation des observations bruitées sans prise en compte des contraintes spatiales (FIG. 5) et avec prise en compte des contraintes (FIG. 7) tendent vers la même qualité vers les observations créées (FIG. 6). Les deux types de variété sont bien reproduits, ainsi que les tendances par parcelle. La prise en compte des propriétés spatiales n'améliore pas et ne dégrade pas non plus de façon évidente les résultats d'assimilation, tout au moins dans la visualisation globale des LAI par pixels, parcelle et variété.

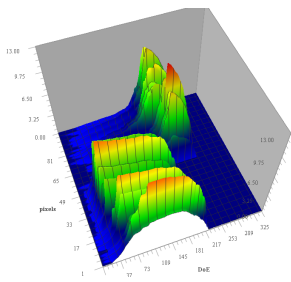


FIG. 5 – LAI obtenus après assimilation sans contraintes spatiales des LAI bruités. $RMSE=0.39623$

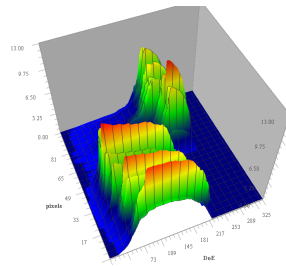


FIG. 6 – LAI observés de référence représentés sans bruit.

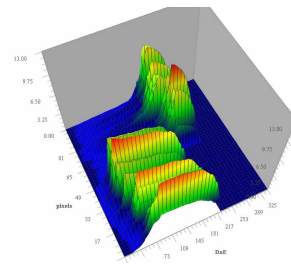


FIG. 7 – LAI obtenus après assimilation avec contraintes spatiales des LAI bruités. $RMSE=0.36573$

Pour évaluer la différence des résultats avec et sans l'aspect spatial, on peut comparer les résultats cumulés des 100 pixels dans les deux cas. Si aucun bruit n'est ajouté aux observations, c'est-à-dire que le modèle est considéré comme parfait, on remarque que les RMSE entre LAI observé et estimé après assimilation sont très satisfaisantes avec les deux méthodes : sans contraintes, on obtient une RMSE sur les 100 pixels de 0.0303, et avec contraintes une RMSE de 0.0173. Cela signifie que le problème est sans doute bien posé dans ces 2 situations, et que le minimiseur retrouve les valeurs de paramètres correspondant parfaitement au minimum de la fonction coût.

Avec les observations bruitées, l'assimilation sans contraintes donne une RMSE de 0.3962, et avec contraintes, une RMSE de 0.3657. Ainsi, l'ajout des contraintes ne permet pas une diminution particulièrement importante de l'écart aux observations (0.03) dans ce cas. Cette méthode obtient une RMSE de 0.366 des LAI, ce qui est plutôt satisfaisant au vu des observations bruitées de 10 %.

La méthode a également été évaluée sur la qualité d'estimation des paramètres d'entrée, puisque l'on connaît les valeurs ayant généré les observations. L'apport des contraintes spatiales en identification de paramètres se dégage particulièrement pour la date de levée relative à la parcelle et les paramètres B et L_{max} relatifs au pixel. Il est difficile de juger de la qualité des estimations des paramètres variétaux, puisque n'ayant considéré que deux variétés, on n'estime que deux valeurs de chaque paramètre, mais on peut affirmer que la contrainte imposée à ces paramètres de rester égaux pour tous les pixels de même variété joue un rôle conséquent et

positif sur l'identification des paramètres relatifs au pixel. D'une manière générale, la méthode stabilise les estimations sur les paramètres auxquels le LAI est peu sensible (leurs estimations pixel par pixel donnent des valeurs très variables, proches de la valeur initiale), et elle améliore les estimations des paramètres sensibles, particulièrement quand les observations sont bruitées. Ainsi, la prise en compte des contraintes a permis de diminuer (faiblement) la RMSE sur le LAI, mais surtout d'identifier certains paramètres plus précisément que lors de l'assimilation classique pixel à pixel.

Enfin, sur le coût numérique de la méthode, la prise en compte des propriétés spatiales permet de diminuer très fortement la quantité de paramètres à identifier, au prix d'une estimation simultanée de ces paramètres. Ceci implique (i) la dérivation du modèle MACRO-BONSAÏ en fonction de 314 paramètres, c'est-à-dire beaucoup plus de sauvegardes de trajectoires (ii) la résolution d'un problème de minimisation de dimension 314 au lieu de 100 problèmes de dimension 6 chacun. L'algorithme de minimisation N1QN3 converge dans cette situation en un nombre très faible d'itérations (25 à 30 en moyenne), soit le même temps que pour estimer 6 paramètres de la méthode initiale.

4 Étude de la sensibilité des résultats d'assimilation à la fréquence temporelle de revisite : intérêt des contraintes spatiales

Les expériences jumelles ont permis d'évaluer l'intérêt de la mise en place des contraintes spatiales sur un cas simple : 23 observations réparties régulièrement sur tout le cycle. Cependant, les observations réelles posent le problème de ne pas être toujours accessibles à l'instant où on le souhaite. Bien que la télédétection permette *a priori* d'acquérir des données régulièrement, il arrive par exemple que de fortes couvertures nuageuses diminuent très fortement la quantité d'observations disponibles. Au vu des résultats précédents, on peut espérer que la méthode avec contraintes sera plus robuste que la méthode classique lors d'une faible quantité d'observations, ou pour un satellite de faible fréquence de revisite.

On propose donc dans cette partie d'étudier divers scénarios de fréquence d'acquisition d'images de télédétection, tout en considérant une certaine probabilité de couverture nuageuse. Les expériences jumelles sont l'outil idéal pour évaluer ces scénarios et juger de l'identifiabilité des paramètres. Les observations simulées et les contraintes spatiales imposées aux paramètres sont les mêmes, seule la quantité d'observations varie, en fonction de :

1. la fréquence de revisite du satellite : passage tous les 3 jours, 7 jours, 15 jours ou 30 jours ;
2. la probabilité de couverture nuageuse (annulant alors l'image), qui est de 0 ou 1 selon une loi uniforme.

Pour chacune des 4 fréquences de revisite du satellite proposées, 1000 probabilités de couverture nuageuse différentes sont appliquées aux observations. Sur ces 1000 simulations, on tend vers une probabilité moyenne de 0.5, et donc vers un nombre total d'observations sur tout le cycle de :

- 58 observations pour une fréquence de 3 jours (entre 0 et 120),
- 24 observations pour 7 jours (entre 0 et 50),

- 11 observations pour 15 jours (entre 0 et 22),
- 6 observations pour un mois (entre 0 et 12).

Les deux méthodes (pixel par pixel et avec contraintes spatiales) ont été testées dans ces 4 cas de fréquence de revisite, sur 1000 simulations de couverture nuageuse, sur les 5 parcelles de deux variétés virtuelles, et avec le même jeu de paramètres initiaux que ceux choisis aléatoirement dans la première partie.

4.1 Apport des contraintes sur l'estimation du LAI

On constate tout d'abord sur le cas médian des 1000 probabilités d'enneuagement que l'assimilation classique devient très instable lorsque l'on considère une fréquence de 15 puis de 30 jours (FIG. 8). En effet, on observe des valeurs irréalistes de LAI sur quelques pixels, où l'algorithme semble ne pas converger. Par contre, l'assimilation prenant en compte les contraintes spatiales reste très homogène jusqu'à la fréquence de 15 jours comprise, et donne un résultat encore satisfaisant par sa stabilité avec une fréquence de 30 jours.

La figure 9 montre l'évolution des RMSE entre LAI observé et estimé selon la fréquence de revisite, avec les deux types d'assimilation. Si pour une fréquence de 3 jours, les deux méthodes aboutissent à une RMSE équivalente de 0.345, dès la fréquence de 7 jours, la RMSE avec l'assimilation classique augmente de 8 % tandis que les contraintes ne font augmenter la RMSE que de 4 %. Avec une fréquence de 15 jours, on augmente la RMSE de 14 % sans contraintes (RMSE=0.57) contre 5% avec contraintes (RMSE=0.44). Enfin, le passage à 30 jours perturbe considérablement les deux méthodes, mais les contraintes obtiennent encore une fois une RMSE inférieure (RMSE=0.71) par rapport à l'assimilation pixel à pixel (RMSE=0.83).

4.2 Apport des contraintes sur l'identification des paramètres

Dans les premiers résultats d'expériences jumelles (§ 3.3), on a constaté que la méthode avec contraintes spatiales améliorait particulièrement l'identification des paramètres. Ces nouvelles expériences vont permettre de préciser cette influence en fonction de la fréquence de revisite du satellite.

Les deux graphiques FIG. 10 représentent la RMSE relative de *l'ensemble des paramètres*, pour chacun des 1000 cas de couverture nuageuse. On constate que l'assimilation pixel à pixel n'est pas tellement sensible à la fréquence puisqu'elle stabilise la RMSE relative sur les paramètres entre 40% et 60%. L'assimilation avec contraintes, elle, augmente sensiblement son erreur quand la fréquence diminue, mais elle reste toujours entre 25% et 30%. Elle est également beaucoup plus stable entre les 1000 répétitions différentes.

Cette évaluation d'ensemble d'erreur sur l'identification des paramètres est limitée puisque le paramètre C , par exemple, est très mal identifié et peut largement influencer la RMSE des 6 paramètres réunis. On propose donc d'observer la RMSE de chaque paramètre sur les 1000 cas de probabilité, pour chaque fréquence de revisite. Avec l'assimilation pixel par pixel, on constate FIG. 11 l'effet important de quelques minima locaux sur la RMSE relative de chaque paramètre. En particulier, la fréquence de 15 jours a subi plus de minima locaux que la fréquence de 30 jours, ce qui ne s'explique que par le choix aléatoire de la probabilité de couverture nuageuse. En moyennant les valeurs estimées du paramètre variétal t_0 sur l'ensemble des pixels de chaque variété, et celles des paramètres parcelaires sur l'ensemble des pixels de

Assimilation d'images sous contraintes spatiales dans un modèle agronomique

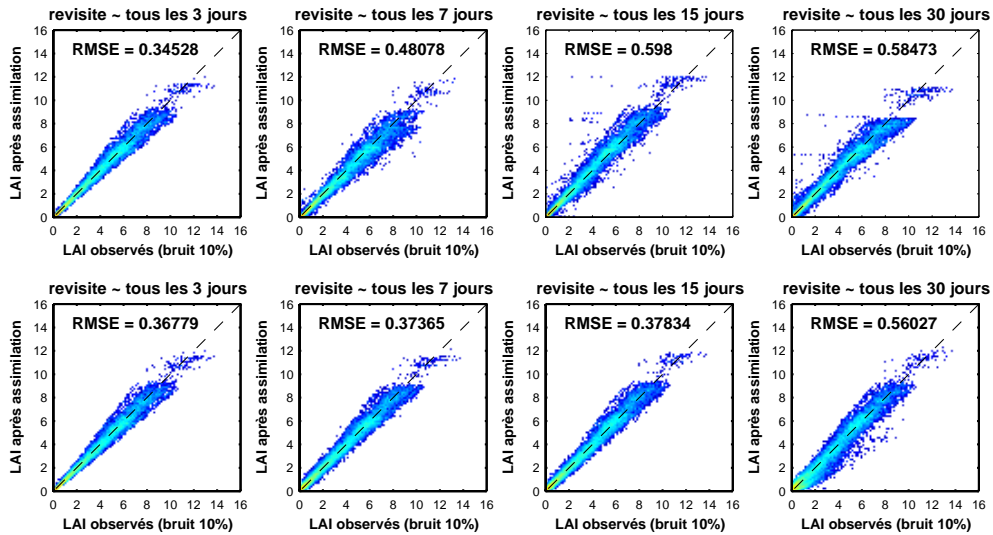


FIG. 8 – Comparaison entre LAI observés et obtenus après assimilation, dans les 4 cas de fréquence de revisite. Représentation en densité apparente (plus clair = plus dense) du cas médian de 1000 expériences de probabilités de nuages différentes. Haut : AD pixel à pixel, bas : AD avec contraintes spatiales.

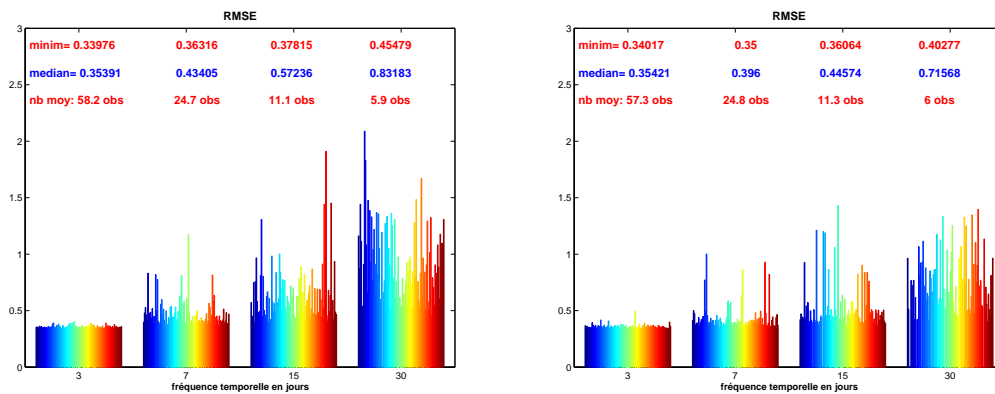


FIG. 9 – Distributions des RMSE entre valeurs de LAI observées et obtenues après assimilation classique (à gauche) et avec contraintes spatiales (à droite) pour les 4 cas de fréquence de revisite, sur 1000 simulations de probabilités de nuages.

chaque parcelle, on minimise l'effet des minima locaux sur les estimations de T_i et ΔT_s . Les résultats de l'assimilation pixel par pixel avec ces moyennes (FIG. 12) sont ainsi comparables à ceux de l'assimilation utilisant les contraintes spatiales (FIG. 13).

Outre l'influence du nombre d'observations pour les deux méthodes sur la RMSE, qui reste toujours inférieure avec les contraintes, ces figures mettent en évidence quelques résultats particuliers aux paramètres. L'estimation de la date de levée t_0 est très mauvaise sans contraintes (RMSE relative de plus de 200%) et plutôt satisfaisante avec les contraintes (RMSE relative de moins de 40% pour les trois premières fréquences). On remarque également que les identifications des paramètres variétaux T_i et ΔT_s sont presque parfaites (au sens de la RRMSE) avec la prise en compte des contraintes pour les trois premières fréquences. Le paramètre le plus influent L_{max} est nettement mieux estimé dès la fréquence temporelle de 7 jours avec les contraintes. Enfin, on constate que le passage de la fréquence à 30 jours ajoute une erreur relative d'environ 10% pour l'assimilation avec contraintes, tandis que les trois premières fréquences restent assez stables.

On peut déduire de ces résultats :

- qu'une fréquence de 7 jours est la limite pour assurer une certaine stabilité et une bonne estimation du LAI et de la plupart des paramètres avec de l'assimilation pixel par pixel. Au-delà de 7 jours, l'algorithme convergera dans de nombreux minima locaux.
- qu'une fréquence de 15 jours associée à la prise en compte de contraintes spatiales garantit une RMSE sur le LAI et une estimation des paramètres presque aussi satisfaisante qu'avec une fréquence de 7 jours. Au-delà de 15 jours, l'algorithme obtiendra des résultats moins satisfaisants mais toujours très stables.

5 Conclusion

A partir de cette première étude de faisabilité, on montre que les contraintes spatiales n'améliorent que marginalement la trajectoire du LAI par rapport à une méthode classique lorsque l'on assimile beaucoup d'observations. Lors d'un plus faible nombre d'observations disponibles, les contraintes spatiales fournissent des résultats beaucoup plus stables face au changement de fréquence de revisite, confirmant la robustesse de la méthode par rapport à l'assimilation pixel par pixel. De plus, cette nouvelle méthode nécessite la minimisation d'une seule fonction coût, plus facile à contrôler lors de divergence ou de minimum local. Enfin, l'identification des paramètres du modèle, qui correspond en général à l'objectif principal de l'assimilation de données, dépend beaucoup moins du nombre d'observations assimilées avec la prise en compte de contraintes spatiales.

On pourrait cependant critiquer le choix déterminé *a priori* des niveaux des paramètres. En effet, BONSAÏ est un modèle semi-mécaniste, dépendant parfois de paramètres dont la signification n'est pas toujours très claire d'un point de vue physiologique : l'intérêt des contraintes spatiales avec ce type de modèle reste assez limité. Si les contraintes spatiales étaient appliquées sur un modèle mécaniste plus complexe, dont les paramètres ont une signification plus précise au niveau des processus, le choix des dépendances spatiales se justifierait alors nettement plus objectivement qu'une décision *a priori*. De plus, ce modèle dépendrait d'une quantité de paramètres bien plus importante que BONSAÏ, donc l'ajout de contraintes réduirait d'autant plus la taille du problème et représenterait un avantage encore plus évident que sur le modèle BONSAÏ.

Assimilation d'images sous contraintes spatiales dans un modèle agronomique

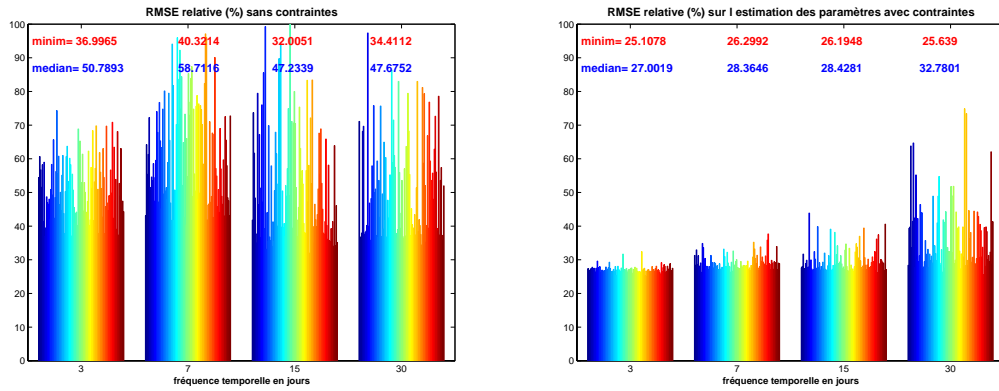


FIG. 10 – Distributions des RMSE relatives en % des paramètres par assimilation **classique** et **avec** contraintes spatiales pour les 4 cas de fréquence de revisite (en abscisse), dans 1000 cas de probabilités de nuages. Les RMSE sont calculées sur les 6 paramètres.

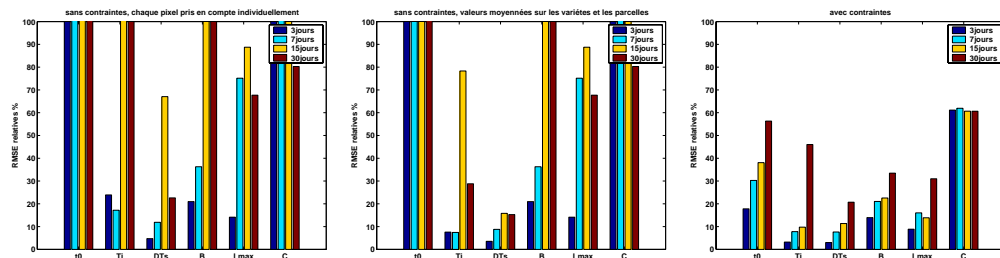


FIG. 11 – Assimilation **classique**. Les valeurs estimées de chaque paramètre sont considérées pixel par pixel.

FIG. 12 – Assimilation **classique**. Les valeurs estimées de t_0 sont moyennées sur les variétés et celles de T_i , ΔT_s sur les parcelles.

FIG. 13 – Assimilation **avec** contraintes spatiales. Les valeurs estimées de t_0 sont fixes par variété et celles de T_i , ΔT_s fixes par parcelle.

FIG. 11-12-13 : RMSE relative (en %), sur 1000 cas de probabilités de nuages, pour les 4 cas de fréquence de revisite représentées chacune par une barre de couleur différente. Les RMSE sont calculées pour chaque paramètre (en abscisse).

De nombreuses perspectives à cette première étude sont envisageables : appliquer cette méthode sur un modèle plus complexe, sur des données réelles, exploiter d'autres informations provenant des données image, telles que la continuité temporelle des images qui peut donner d'autres informations, la prise en compte des corrélations spatiales par image : entre les données pixels, par exemple en rapport à leur distance ou connectivité les uns aux autres, et les erreurs associées, notamment si ce sont des pseudo-observations, provenant d'une inversion de modèle de transfert radiatif par exemple.

Références

- Baret, F. (1986). *Contribution au suivi radiométrique de cultures de céréales*. Ph. D. thesis, Université d'Orsay.
- Brisson, N. e. a. (1998). STICS : a generic model for the simulation of crops and their water and nitrogen balances. I. theory and parameterisation applied to wheat and maize. *Agronomie 18*, 311–346.
- Brisson, N. e. a. (2002). STICS : a generic model for simulating crops and their water and nitrogen balances. II. model validation for wheat and maize. *Agronomie 22*, 69–92.
- Gilbert, J.-C. et C. Lemaréchal (1989). Some numerical experiments with variable storage quasi-newton algorithms. *Math. Prog. 45*, 407–435.
- Hascoët, L., S. Fidanova, et C. Held (2001). Adjoining independent computations. In G. Corliss, C. Faure, A. Griewank, L. Hascoët, et U. Naumann (Eds.), *Automatic Differentiation of Algorithms : From Simulation to Optimization*, Computer and Information Science, Chapter 35, pp. 299–304. New York, NY : Springer.
- Lauvernet, C., F. Baret, L. Hascoët, S. Buis, et F.-X. Le Dimet (2008). Multitemporal-patch ensemble inversion of coupled surface-atmosphere radiative transfer models for land surface characterization. *Remote Sens. Environ. 112*(3), 851–861.
- Le Dimet, F.-X. et O. Talagrand (1986). Variational algorithms for analysis and assimilation of meteorological observations : Theoretical aspects. *Tellus 38A*, 97–110.

Summary

Information contained in time series of image data should be explicitly exploited in data assimilation methods instead of operating over single pixels. This study proposes to adapt a variational data assimilation method of LAI (Leaf Area Index) images in a crop model. The method assumes that the parameters are governed spatially at some levels (cultivar, field, and pixel), while some of them are assumed to be stable temporally over the whole image. Such constraints help at reducing the size of the inverse problem, transforming the usual assimilation scheme into simultaneous pixel patterns. DA with constraints is applied to the semi-mechanistic model BONSAÏ and evaluated by twin experiments both on the quality of LAI prediction and on parameter estimates. Sensitivity to the observations frequency is also evaluated. The constraints improve the method's robustness and estimates when the number of observations available decreases, compared to the conventional method.

Assimilation de données hétérogènes pour des systèmes faiblement couplés. Application au problème inverse de l'électrocardiologie

Philippe Moireau*, Cesare Corrado**,***
Jean-Frédéric Gerbeau**,***

*Inria Saclay-Ile de France, Team M Ξ DISIM,
Campus de l'Ecole Polytechnique, 91128 Palaiseau

**Inria Paris-Rocquencourt, Project-Team Reo,
BP 105, Le Chesnay Cedex

***UPMC Paris 6, Laboratoire Jacques-Louis Lions

Résumé. Ce travail considère le problème inverse de l'électrocardiologie, c'est-à-dire l'inversion des mesures d'électrocardiogramme (ECG), d'un point de vue original. Ce problème inverse initialement mal posé est intégré à un problème d'assimilation de données plus globale en lui adjoignant des données supplémentaires issues de l'imagerie dynamique cardiaque. En considérant alors le problème couplé de l'électrophysiologie et de la mécanique cardiaque en résultant nous montrons que l'estimation est efficace pour l'état et les paramètres. En pratique, nous proposons une méthode d'assimilation de données séquentielle combinant un estimateur de Luenberger sur une partie mécanique et un estimateur optimal réduit à l'état électrique et aux paramètres.

1 Introduction

Depuis quelques années, il existe un intérêt croissant pour l'assimilation de données associées à des modèles biophysiques pour des applications médicales, notamment en cardiologie. L'objectif est clairement d'offrir des nouveaux outils de diagnostic par la confrontation modèle-données. Il est vrai que des données de nature très différente sont aujourd'hui accessibles. On pense typiquement à l'imagerie médicale, aux mesures de pression par cathérisation ou de flux sanguin par Doppler, ou encore des électrocardiogrammes (ECG). Jusqu'à maintenant les tentatives d'inversion de ces données pour caractériser l'état d'un patient se sont concentrées sur un seul type de données choisi, combiné au modèle physique qui lui est associé. Par exemple, on a cherché à caractériser l'état mécanique du coeur à travers des données d'imagerie du mouvement de l'organe (Moireau et al., 2009), ou plus encore on a cherché à définir l'état électrophysiologique d'un patient par l'inversion des données ECG (Boulakia et al., 2008). Ce dernier problème appelé "problème inverse de l'électrocardiologie" est réputé pour être particulièrement mal posé car il revient à inverser un opérateur de diffusion de l'état électrique du coeur jusqu'au thorax du patient. A notre connaissance, personne n'a pour l'instant considéré les données hétérogènes électriques et mécaniques dans leur ensemble, combinées à

la nature profondément multiphysique des phénomènes régissant la contraction cardiaque. Or si l'inversion des seules données ECG est mal posée, que dire par contre de l'inversion de données ECG combinées à des données d'imagerie médicale montrant la réponse mécanique de l'activation électrique excitant le tissu cardiaque à chaque battement ? Evidemment, combiner ces données hétérogènes nécessite des modèles eux-aussi multiphysiques. Nous considérerons donc un exemple de modèle d'électrophysiologie couplé à un modèle de contraction cardiaque. Chaque modèle, électrique et mécanique correspond à l'état de l'art en modélisation cardiaque. Le couplage, lui, sera supposé faible dans le sens où on néglige l'influence de la mécanique vers l'électrophysiologie pour ne se focaliser que sur l'influence de l'électrophysiologie vers la mécanique. Profitant de la structure particulière du couplage faible nous proposons une méthode d'assimilation de données séquentielle efficace combinant des filtres optimaux et des filtres de Luenberger. Nous montrons alors que certains paramètres qui étaient mal estimés via le modèle électrophysiologique seul sont désormais identifiables.

2 Principes d'assimilation de données

L'assimilation de données se formule généralement dans un contexte très général soit stochastique soit déterministe. Nous limiterons ici nos rappels au cadre déterministe en considérant un système dynamique de la forme

$$\dot{X} = A(X, \theta, t). \quad (1)$$

où dans cette équation, X décrit l'état du système, c'est-à-dire la concaténation des quantités physiques décrivant l'évolution en temps du système. Le modèle, lui, est résumé par l'opérateur dynamique A qui s'applique à l'état et dépend d'un certain nombre de paramètres à identifier que nous agrégeons dans θ . Dans cette formulation, la condition initiale et les paramètres doivent être fixés et il est classique de les décomposer en des composantes connues X_\diamond et θ_\diamond appelées *a priori*, et des composantes à estimer ζ^x et ζ^θ :

$$X(0) = X_\diamond + \zeta^x, \quad \theta = \theta_\diamond + \zeta^\theta.$$

généralisant une solution $X_{[\zeta^x, \zeta^\theta]}$. Dans cet exposé, nous nous limiterons à ces deux types d'incertitudes bien qu'en général des erreurs non paramétriques sur modèles puissent aussi intervenir. De plus, ces deux erreurs peuvent être unifiées sous un même cadre formel en considérant l'état augmenté ${}^bX = \begin{pmatrix} X \\ \theta \end{pmatrix}$ de dynamique

$${}^b\dot{X} = {}^bA({}^bX), \quad \text{et } {}^bX(0) = \begin{pmatrix} X^{(0)} \\ \theta \end{pmatrix} = X_\diamond + \zeta. \quad (2)$$

En regard du système existent des mesures, ou observations, disponibles en temps comme une vision incomplète de la trajectoire cible du système X^\dagger . Nous les représentons sous la forme

$$Z = H(X^\dagger, t) + \chi(t), \quad (3)$$

où Z désigne les données, H est l'opérateur d'observation et χ est un bruit de mesure.

Enfin nous limitons notre description formelle au cadre de la dimension finie. Pour les problèmes physiques modélisés par des équations aux dérivées partielles, ceci revient à d'abord considérer une première étape de discrétisation spatiale.

2.1 Stratégie par filtrage

En assimilation de données, on distingue généralement deux approches. La première est dite variationnelle et fut popularisée par le 4D-Var (Blum et al., 2008). Elle consiste à minimiser un critère moindres carrés du type

$$\mathcal{J}(\zeta) = \frac{1}{2} \|\zeta\|_{P_\diamond}^2 + \frac{1}{2} \int_0^T \|Z - H({}^bX_{[\zeta]})\|_M^2 dt, \quad (4)$$

où M est une métrique sur l'erreur d'observation, P_\diamond est l'inverse d'une métrique interprétable sous la forme d'une covariance d'incertitude de condition initiale. Le 4D-Var consiste à minimiser \mathcal{J} sous contrainte de la dynamique du modèle. La minimisation sous contrainte s'effectue généralement par l'introduction d'une variable adjointe. Le critère est alors minimisé par une méthode de descente de gradient.

Quand on ne cherche pas nécessairement à estimer la condition initiale du système, mais simplement à identifier le système pour toute erreur possible de condition initiale, une approche séquentielle peut aussi être envisagée. C'est la deuxième catégorie d'approche. Le principe est d'introduire un système modifié appelé estimateur, ou *observateur*, qui incorpore une correction fondée sur la dissimilarité entre la trajectoire calculée et les mesures fournies. Le nouveau système s'écrit alors

$${}^b\dot{\hat{X}} = A({}^b\hat{X}, t) + G(Z - H({}^b\hat{X})), \quad {}^b\hat{X} = {}^bX_\diamond, \quad (5)$$

où G est appelé l'opérateur de filtrage, ou opérateur de gain, caractérisant le contrôle que nous appliquons au système en fonction de la dissimilarité calculée par $Z - H({}^b\hat{X})$. Là encore, on obtient après réécriture sur l'état et les paramètres :

$$\begin{cases} \dot{\hat{X}} = A(\hat{X}, \hat{\theta}, t) + G^x(Z - H(\hat{X})), & \hat{X} = X_\diamond, \\ \dot{\hat{\theta}} = G^\theta(Z - H(\hat{X})), & \hat{\theta}(0) = \theta_\diamond. \end{cases} \quad (6)$$

La question est donc en filtrage de définir un opérateur de gain efficace. Au même titre que le 4D-Var peut s'appliquer à n'importe quel système, une classe essentielle d'opérateurs de filtrage, appelés opérateurs de gains optimaux peuvent en fait se calculer pour tous les systèmes physiques. L'idée est de construire un opérateur de gain à partir du problème variationnel en définissant l'estimateur optimal correspondant par

$${}^b\hat{X}(t) = {}^bX_{[\operatorname{argmin} \mathcal{J}(\cdot, t)]}(t).$$

Le filtre apparaît alors naturellement en dérivant cette définition par rapport au temps qui intervient à la fois dans la dynamique et comme temps final du critère. Lorsque les opérateurs de modèle et d'observation sont linéaires on retrouve alors le fameux filtre de Kalman (Simon, 2006). Dans ce cas, l'opérateur de gain est obtenu à partir de la résolution d'une équation de Riccati $\dot{P} = {}^bAP + P{}^bA^\top - PH^\top MHP$ sur un opérateur P , dit de *covariance*. Dans le cas non-linéaire, le gain optimal est, cette fois, obtenu en résolvant une équation de Hamilton-Jacobi-Bellman (HJB) sur l'espace des états (Simon, 2006). La complexité de la résolution de cette équation de HJB a conduit de nombreux auteurs à recourir alors à une approximation sous-optimale en résolvant au contraire une équation de Riccati à partir des tangents des

opérateurs de modèle et de d'observation. C'est le fameux filtre de Kalman étendu (EKF) :

$$\dot{P} = (d^b A)P + P(d^b A)^\top - P(dH)^\top M(dH)P. \quad (7)$$

Il faut noter qu'il existe aussi d'autres variantes déterministes autour des approximations sous-optimales. On peut citer par exemple la stratégie «Unscented Kalman Filter (UKF)» (Julier, 2002) où, après discrétisation en temps, on remplace les opérateurs tangents calculés dans EKF par des interpolations sur des états définis, *i.e.* des particules appelées sigma-points. Quoi qu'il en soit, même en ayant recours à une stratégie approchée, les filtrages optimaux souffrent d'un défaut récurrent lié au calcul des opérateurs de covariance qui sont des opérateurs «pleins». Ainsi pour des systèmes de grande dimension typiquement issus de la discrétisation d'EDP, ces opérateurs de covariance, et donc les opérateurs de gain associés, sont incalculables en pratique.

2.1.1 Au delà de la malédiction de la dimensionalité

Pour contrer l'impossibilité de calculer les filtres optimaux en pratique, il existe deux grandes stratégies : le filtrage optimale de *rang réduit*, et les filtres dit de *Luenberger*.

Filtrage de rang réduit Le filtrage optimal de rang réduit se fonde sur une réduction de l'opérateur de covariance du filtrage optimal idéalement stable au cours du temps. Typiquement, on considère l'*ansatz*

$$\forall t, \quad P(t) = L(t)U(t)^{-1}L(t)^\top, \quad (8)$$

où U est un opérateur inversible de rang réduit r et L un opérateur d'extension de l'espace réduit vers l'espace initial (L^\top est donc l'opérateur de projection dual). On montre alors (Moireau et al., 2008), pour des opérateurs linéaires au moins, que l'équation de Riccati (7) se réduit à résoudre les deux dynamiques

$$\dot{L} = {}^b A L \text{ et } \dot{U} = L^\top H^\top M H L, \quad (9)$$

qui sont désormais calculables en pratique. En non-linéaire, on procède alors par analogie en proposant le filtre de Kalman étendu d'ordre réduit (RoEKF) sous la forme

$$\dot{L} = (d^b A)L \text{ et } \dot{U} = L^\top (dH)^\top M (dH)L. \quad (10)$$

Un cas particulier intéressant concerne le cas où la réduction s'opère sur l'espace paramétrique (Moireau et al., 2008). On obtient alors

$$\begin{cases} \dot{\hat{X}} = A(\hat{X}, \hat{\theta}) + L^x \dot{\hat{\theta}}, & \hat{X}(0) = X_\diamond, \\ \dot{\hat{\theta}} = U^{-1} L^{x^\top} (d_x H)^\top M (Z - H(\hat{X})), & \hat{\theta}(0) = \theta_\diamond, \\ \dot{L}^x = (d_x A)L^x + d_\theta A, & L^x(0) = \mathbb{1}, \\ \dot{U} = L^{x^\top} (d_x H)^\top M (d_x H)L^x, & U(0) = U_\diamond. \end{cases} \quad (11)$$

Enfin, la réduction peut aussi s'opérer à partir du filtre UKF conduisant alors à un estimateur (RoUKF) similaire à (10) où les opérateurs tangents sont simplement remplacés par des opérateurs interpolés à partir de particules (Moireau et Chapelle, 2011).

Filtrage de Luenberger Une stratégie très différente apparaît avec les observateurs de Luenberger. L'idée n'est plus de s'appuyer sur un principe variationnel sous-jacent mais simplement de définir l'opérateur de gain tel que l'erreur d'estimation tende vers 0 au cours du temps. Considérons la solution cible ${}^bX^\dagger$, on définit l'erreur d'estimation ${}^b\tilde{X} = {}^bX^\dagger - {}^b\hat{X}$ dont on souhaite ${}^b\tilde{X}(t) \xrightarrow{t \rightarrow \infty} 0$.

Ainsi l'objectif est de définir un opérateur de contrôle-*feedback* tel que l'erreur d'estimation soit stabilisée à 0. *A minima* on souhaitera que pour des erreurs initiales faibles le système de l'erreur linéarisée autour de 0, notée ${}^b\delta\tilde{X}$ et de dynamique

$${}^b\delta\tilde{X} = (dA + GdH){}^b\delta\tilde{X}, \quad {}^b\delta\tilde{X}(0) = \zeta, \quad (12)$$

soit stable en temps.

Il existe pour chaque type de systèmes physiques une littérature fournie concernant la définition de contrôleurs simples. Cette approche, connue aussi sous le nom de *nudging* en assimilation de données, a produit un ensemble de filtres calculables pour des équations aussi diverses que les équations de transport (Auroux et Blum, 2005), les ondes, la mécanique des fluides ou solides (Bertoglio et al., 2011) etc.

À noter par ailleurs que les opérateurs optimaux vérifient aussi une propriété de convergence de l'estimateur, y compris pour les versions approchées (Simon, 2006).

3 Assimilation de données en cardiologie

Nous pouvons maintenant appliquer ces fondements de l'assimilation de données à l'électro-mécanique cardiaque. Nous commençons par considérer l'électrophysiologie cardiaque indépendamment de sa résultante mécanique. Dans un deuxième temps, nous considérons un modèle mécanique de contraction répondant à une entrée électrique donnée. Nous aborderons dans la dernière section de ce travail les apports obtenus en considérant le couplage dans son ensemble.

3.1 Estimation en électrophysiologie

Un modèle d'électrophysiologie Nous nous appuyons ici sur l'état de l'art en électrophysiologie en considérant le modèle bidomaine qui régit la propagation de la dépolarisation électrique à la surface du coeur (Boulakia et al., 2008; Chapelle et al., 2009). Soit le système d'EDP posé en tout point du domaine «coeur» Ω_0^H :

$$\begin{cases} a_m (c_m \partial_t v_m + I_{\text{ion}}(v_m, w)) - \nabla \cdot (\underline{D}_i \cdot \nabla (u_e + v_m)) = a_m I_{\text{app}}, & \text{dans } \Omega_0^H, \\ -\nabla \cdot ((\underline{D}_i + \underline{D}_e) \cdot \nabla u_e) - \nabla \cdot (\underline{D}_i \cdot \nabla v_m) = 0, & \text{dans } \Omega_0^H, \\ \partial_t w + g(v_m, w) = 0, & \text{dans } \Omega_0^H, \\ (\underline{D}_e \cdot \nabla u_e) \cdot \underline{n} = 0, & \text{sur } \partial\Omega_0^H, \\ (\underline{D}_i \cdot \nabla (u_e + v_m)) \cdot \underline{n} = 0, & \text{sur } \partial\Omega_0^H. \end{cases} \quad (13)$$

où v_m représente le potentiel transmembranaire, u_e le potentiel extra-cellulaire, c_m est la capacitance de la membrane par unité d'aire, a_m est le ratio de surface cellulaire par unité de

volume, $\underline{D}_i, \underline{D}_e$ sont les tenseurs de conductivité intra- et extra-cellulaire et I_{app} est un courant extérieur appliqué par unité de volume. Dans (13), la fonction g représente un modèle ionique, c'est-à-dire un ensemble d'équations différentielles définissant l'évolution locale des ions à la surface de la membrane lors de la dépolarisation. Plusieurs choix sont possibles dans la littérature et nous choisissons ici le modèle de Mitchell-Schaeffer de complexité modérée :

$$\begin{cases} I_{\text{ion}}(v_m, w) = -\frac{w}{\tau_{\text{in}}} \frac{(v_m - V_{\text{min}})^2 (V_{\text{max}} - v_m)}{V_{\text{max}} - V_{\text{min}}} + \frac{1}{\tau_{\text{out}}} \frac{v_m - V_{\text{min}}}{V_{\text{max}} - V_{\text{min}}}, \\ g(v_m, w) = \begin{cases} \frac{w}{\tau_{\text{open}}} - \frac{1}{\tau_{\text{open}} (V_{\text{max}} - V_{\text{min}})^2} & \text{if } v_m \leq V_{\text{gate}}, \\ \frac{w}{\tau_{\text{close}}} & \text{if } v_m > V_{\text{gate}}, \end{cases} \end{cases} \quad (14)$$

où $V_{\text{gate}}, \tau_{\text{in}}, \tau_{\text{out}}, \tau_{\text{open}}, \tau_{\text{close}}$ sont des paramètres associés à l'ouverture de canaux ioniques permettant le passage des ions à travers la membrane cellulaire et V_{min} et V_{max} sont des constantes de mise à l'échelle (typiquement $V_{\text{min}} = -80mV$ et $V_{\text{max}} = 20mV$).

Après discrétisation en éléments finis, ce modèle se présente donc sous la forme

$$\begin{cases} \mathbf{M}_a^e \dot{\vec{V}}_m + \mathbf{K}_i^e (\vec{V}_m + \vec{U}_e) = \mathbf{M}_I^e (\vec{I}_{\text{app}} - \vec{I}_{\text{ion}}(\vec{V}_m, \vec{W})), \\ (\mathbf{K}_i^e + \mathbf{K}_e^e) \vec{U}_e + \mathbf{K}_i^e \vec{V}_m = 0, \\ \dot{\vec{W}} + \vec{G}(\vec{V}_m, \vec{W}) = 0, \end{cases} \quad (15)$$

où les opérateurs $\mathbf{M}_a^e, \mathbf{M}_I^e$, sont des opérateurs de type masse et \mathbf{K}_i^e et \mathbf{K}_e^e des opérateurs de « raideur » associés aux conductivités \underline{D}_i et \underline{D}_e . Finalement en formalisme d'état, on écrit

$$\begin{cases} \dot{X}^e = A^e(X^e, \theta^e), \\ X^e(0) = X_\diamond^e + \zeta_{X^e}, \\ \theta^e = \theta_\diamond^e + \zeta_{\theta^e}, \end{cases} \quad (16)$$

où l'état est $X^e = \begin{pmatrix} \vec{V}_m \\ \vec{W} \end{pmatrix}$ tandis que \vec{U}_e apparaît comme une variable auxiliaire vérifiant l'équilibre quasi-statique défini dans (15)₂.

Les mesures ECG Les mesures les plus communément disponibles de l'activité électrique sont les électrocardiogrammes (ECG) qui évaluent des différences de potentiel électrique à la surface du thorax Ω^T . Ainsi les ECG mesurent l'image, à la surface du thorax, de la dépolarisation électrique cardiaque simplement diffusée dans le reste de l'organisme :

$$\begin{cases} \nabla \cdot (\underline{D}_T \cdot \nabla u_T) = 0, & \text{dans } \Omega^T \\ u_T = u_e, & \text{sur } \partial\Omega_0^H \\ (\underline{D}_T \cdot \nabla u_T) \cdot \underline{n} = 0, & \text{sur } \partial\Omega^T \setminus \partial\Omega_0^H \end{cases} \quad (17)$$

Ce modèle de diffusion conduit donc à la définition d'un opérateur d'observation linéaire

$$Z^e(t) = H^e X^e + \chi^e, \quad (18)$$

où χ^e prend en compte les erreurs de mesure et la discrétisation. Cet opérateur d'observation H^e n'est clairement pas inversible puisqu'il est associé à un phénomène de diffusion. Ce n'est qu'intégré en temps que nous avons espoir de l'inverser pour reconstruire l'état de dépolarisation du coeur et potentiellement identifier certains paramètres constitutifs du modèle bidomaine.

Estimation par filtrage optimal réduit Sur ce modèle et ces données, nous proposons de définir un estimateur séquentiel de rang réduit. Deux réductions sont potentiellement envisageables. Réduire l'incertitude initiale à un certain nombre de paramètres du modèle, ou inclure une partie de l'état dans l'incertitude en définissant ses composantes principales par le biais d'un modèle réduit. Notre choix se porte alors sur une décomposition de l'état en considérant une base POD (Proper Orthogonal Decomposition) où on extrait, d'un ensemble de trajectoires déjà calculées, les composantes dominantes de l'état (Boulakia et al., 2012; Chapelle et al., 2013). Ces composantes doivent ici être calculées à partir de trajectoires qui prennent en compte les variations paramétriques incertaines (Chapelle et al., 2013). Autrement dit on calcule un ensemble de trajectoires dans un intervalle paramétrique raisonnable à partir desquelles on calcule les composantes dominantes par recherche de valeurs singulières (SVD).

En pratique, on désigne par $\Pi \in \mathbb{M}_{N^e, N_{\text{POD}}}(\mathbb{R})$ la matrice des N_{POD} premiers modes POD. Ces modes sont orthogonaux par rapport à un produit scalaire $l^2(\mathbb{R}^{N_{\text{POD}}})$ ou $L^2(\Omega^H)$ de matrice grammienne associée M_{POD} . Les coefficients POD X^e de notre décomposition sont donnés par $\alpha = \Pi^T M_{\text{POD}} X^e$. Définissant $X_{\text{POD}}^e = \Pi \alpha$ et $X_{\perp}^e = X^e - \Pi \alpha$, le vecteur d'état se décompose en $X^e = X_{\text{POD}}^e + X_{\perp}^e$. On applique alors un filtrage optimal de rang réduit à

$$X^r = \begin{pmatrix} \alpha \\ \theta \end{pmatrix} \in \mathbb{R}^r = \mathbb{R}^{N_{\text{POD}}+d} \text{ tel que } X^e = \begin{pmatrix} X_{\perp}^e \\ X^r \end{pmatrix}. \quad (19)$$

3.2 Estimation en mécanique cardiaque

Un modèle mécanique Comme pour l'électrophysiologie nous considérons un exemple de modèle de contraction cardiaque représentatif de l'état de l'art et proposé dans (Chapelle et al., 2009; Chabiniok et al., 2011). Ce modèle définit l'évolution Lagrangienne du déplacement \underline{y} du tissu cardiaque au cours du temps :

$$\begin{cases} \partial_t \underline{y} = \underline{v}, & \text{dans } \Omega_0^H \\ \rho \partial_t \underline{v} - \nabla \cdot (\underline{T}, V_m) = 0, & \text{dans } \Omega_0^H \\ \underline{T} \cdot \underline{n} = k_s \underline{y} + c_s \underline{v}, & \text{sur } \Gamma_n \\ \underline{T} \cdot \underline{n} = J p_{v,i} \underline{F}^{-T} \cdot \underline{n}, & \text{sur } \Gamma_{c,i} \\ \underline{T} \cdot \underline{n} = 0, & \text{sur } \partial \Omega_0^H \setminus ((\cup_i \Gamma_{c,i}) \cap \Gamma_n) \end{cases} \quad (20)$$

où \underline{v} est la vitesse, ρ est la densité et \underline{T} est le premier tenseur des contraintes de Piola-Kirchhoff qui est une fonction de la déformation et du potentiel transmembranaire via le modèle défini dans Chapelle et al. (2012). Concernant les conditions aux limites, k_s et c_s sont des paramètres de conditions de rappels sur une zone Γ_n de l'épicarde, $p_{v,i}$ sont les pressions intracavitaires s'appliquant sur chaque endocarde, \underline{F} est le gradient de la déformation et J son déterminant.

Après discrétisation, on obtient donc

$$\begin{cases} \dot{\vec{Y}} = \vec{V}, \\ \mathbf{M}^m \dot{\vec{V}} + \vec{\mathbf{K}}^m(\vec{Y}, \vec{V}, \vec{l}_c) + \mathbf{M}_{k_s, \Gamma_n}^m \vec{Y} + \mathbf{M}_{c_s, \Gamma_n}^m \vec{V} = - \sum_i p_{v,i} \vec{\mathbf{N}}_i^m(\vec{Y}), \\ \dot{\vec{l}}_c = \vec{\mathbf{B}}^m(\vec{l}_c, \vec{Y}, \vec{V}). \end{cases} \quad (21)$$

où \mathbf{M}^m est la matrice de masse, $\vec{\mathbf{K}}^m$ est la raideur non-linéaire. D'autre part $\mathbf{M}_{k_s, \Gamma_n}^m$ et $\mathbf{M}_{c_s, \Gamma_n}^m$ sont les matrices prenant en compte les conditions aux limites associées à k_s et c_s et $\vec{\mathbf{N}}_i^m(\vec{Y})$ est l'opérateur de pression suivieuse associé à chaque endocarde. La réécriture en fonction de l'état $X^m = \begin{pmatrix} \vec{Y} \\ \vec{V} \\ \vec{l}_c \end{pmatrix}$ donne finalement

$$\begin{cases} \dot{X}^m = \mathbf{A}^m(X^m, X^e, \theta^m), \\ X^m(0) = X_{\diamond}^m + \zeta_{X^m}, \\ \theta^m = \theta_{\diamond}^m + \zeta_{\theta^m}. \end{cases} \quad (22)$$

Des mesures cinématiques Les mesures de la déformation du myocarde, sont typiquement obtenues par post-traitement des données d'imagerie médicale par exemple IRM (Chabiniok et al., 2011). Suivant les protocoles d'imagerie, des données plus ou moins riches sont accessibles. Une modalité très puissante, bien qu'encore peu répandue en routine clinique est l'IRM de marquage tissulaire où il est alors possible de suivre les déplacements de points matériels du tissu marqués magnétiquement. Dans ce cas, on retrouve un opérateur d'observation linéaire

$$Z^m(t) = \mathbf{H}^m X^m + \chi^m,$$

où χ^m représente le bruit de mesure, de post-traitement et de discrétisation.

Nous nous limiterons dans cet article à ce type de données, mais des extensions à des données beaucoup plus complexes sont possibles. Par exemple dans (Chabiniok et al., 2011), on considère que l'information cinématique est donnée par l'évolution des contours du domaine du myocarde. Ces données de contours sont alors intégrées à une boucle d'assimilation de données en calculant en chaque point du contour simulé la distance au contour mesuré.

Estimation par filtres de Luenberger Sur ce système mécanique nous proposons d'utiliser un filtre de type Luenberger proposé dans Moireau et al. (2009). On définit après discrétisation

$$\begin{cases} \dot{\hat{\vec{Y}}} = \hat{\vec{V}} + \gamma \vec{\text{Ext}}(Z^m - \mathbf{H}^m(X)), \\ \mathbf{M}^m \dot{\hat{\vec{V}}} + \vec{\mathbf{K}}^m(\hat{\vec{Y}}, \hat{\vec{V}}) + \mathbf{M}_{k_s, \Gamma_n}^m \hat{\vec{Y}} + \mathbf{M}_{c_s, \Gamma_n}^m \hat{\vec{V}} = \sum_i p_{v,i} \vec{\mathbf{N}}_i^m(\hat{\vec{Y}}), \end{cases} \quad (23)$$

où $\vec{\text{Ext}}$ est un opérateur d'extension qui reconstruit une donnée sur Ω_0^H à partir de la donnée observée en résolvant le problème statique associé à (20). L'étude de ce filtre a été menée en détails dans Moireau et al. (2009) pour des systèmes elastodynamiques linéaires puis son efficacité a été démontrée numériquement dans le cas d'un modèle cardiaque (Chabiniok et al.,

2011). L'originalité de ce filtre vient du fait que, dans l'observateur (23), la relation décrivant la vitesse comme la dérivée en temps du déplacement est modifiée par un terme de feedback sur l'erreur d'observation. Cette modification n'est possible que sous la condition que l'observateur (23) est un observateur virtuel *in silico*. Dans l'espace d'état, ce filtre de Luenberger s'écrit

$$\dot{\hat{X}}^m = A^m(X^m) + G_L^m(Z^m - H^m(X^m)), \quad (24)$$

où G_L^m est le filtre de Luenberger.

Extension à l'estimation conjointe état-paramètres Nous devons désormais aborder la question de l'identification de paramètres potentiels pour la mécanique cardiaque sachant que nous avons décidé d'estimer l'état par un filtre de Luenberger. Pour ce faire, nous nous appuyons sur Moireau et al. (2008) en appliquant, à partir de l'observateur d'état, un filtre réduit sur les paramètres. En effet, l'estimateur d'état a pour objectif fondamental de rattraper les incertitudes sur l'état ; il est donc cohérent de limiter l'utilisation du filtrage optimal de rang réduit aux incertitudes paramétriques restantes. On obtient finalement

$$\begin{cases} \dot{\hat{X}}^m = A^m(\hat{X}^m, \hat{\theta}^m) + G_L^m(Z^m - H^m(\hat{X}^m)), \\ \quad + L^m(U^m)^{-1}(L^m)^\top dH^m M^m(Z^m - H^m(\hat{X}^m)), \quad \hat{X}^m(0) = X_\diamond^m, \\ \dot{\hat{\theta}}^m = L^m(U^m)^{-1}(L^m)^\top (dH^m)^\top M^m(Z^m - H^m(\hat{X}^m)), \quad \hat{\theta}^m(0) = \theta_\diamond^m, \\ \dot{L}^m = (d_x A^m(\hat{X}^m, \hat{\theta}^m) - G_L^m dH^m) L^m + d_\theta A^m(\hat{X}^m, \hat{\theta}^m) \quad L^m = 0, \\ \dot{U}^m = (L^m)^\top (dH^m)^\top M^m (dH^m) L^m, \quad U(0) = U_\diamond^\theta. \end{cases} \quad (25)$$

L'analyse de cet estimateur à été proposée dans Moireau et al. (2008), là encore dans le cadre des équations d'erreurs linéarisées. Enfin de nombreux exemples (non-linéaires) d'applications de cet observateur ont été produits dans la littérature, voir par exemple Chabiniok et al. (2011).

3.3 Estimation de système couplé en électrocardiologie

La dernière question que nous souhaitons aborder est le couplage des deux types d'observateurs pour le système couplé de l'électro-mécanique cardiaque, résumé sous la forme

$$\begin{cases} \dot{X}^e = A^e(X^e), \quad X^e(0) = X_\diamond^e + \zeta^e, \\ \dot{X}^m = A^m(X^m, X^e), \quad X^m(0) = X_\diamond^m + \zeta^m. \end{cases} \quad (26)$$

Nous insistons sur la particularité du couplage étudié ici dit «one way» dans le sens où seul l'électrophysiologie a un impact sur la mécanique mais nous négligeons l'effet de la mécanique sur l'électrophysiologie.

Estimateur couplé La partie paramétrique ne pose aucune difficulté une fois les filtres d'état proprement couplés. Nous nous limiterons donc à présenter la formulation correspondant à l'estimation d'état seul. Nous utiliserons les notations d_m et d_e pour représenter les différentielles partielles par rapport à « X^m et X^e » respectivement, alors que « d » continuera à

Assimilation de données hétérogènes en électrocardiologie

caractériser la différentielle totale. On introduit dans le RoEKF $L = \begin{pmatrix} L^e \\ L^m \end{pmatrix}$ pour finalement obtenir depuis $\hat{X}^e(0) = X_\diamond^e$, $\hat{X}^m(0) = X_\diamond^m$, $L^m = 0$, $L^e = \mathbb{1}$ et $U(0) = U_\diamond^e$ la dynamique

$$\begin{cases} \dot{\hat{X}}^e = A^e(\hat{X}^e) + L^e U^{-1} L^\top dH^\top M(Z - H(\hat{X})), \\ \dot{\hat{X}}^m = A^m(\hat{X}^m, \hat{X}^e) + G_L^m(Z^m - H^m(\hat{X}^m)) + L^m U^{-1} L^\top dH^\top M(Z - H(\hat{X})), \\ \dot{L}^e = (dA^e(\hat{X}^e))L^e, \\ \dot{L}^m = (d_m A^m(\hat{X}^m, \hat{X}^e) - G_L^m dH^m(\hat{X}^m))L^m + (d_e A^m(\hat{X}^m, \hat{X}^e))L^e, \\ \dot{U} = L^\top (dH)^\top M(dH)L. \end{cases} \quad (27)$$

Il faut noter que, dans la dynamique (27), le filtre de Luenberger s'applique uniquement sur la partie mécanique. Cependant le filtre optimal corrige l'évolution électrophysiologique à partir des données d'électrophysiologie et de mécanique. Cette correction est alors répercutée sur la mécanique qui peut se réécrire sous la forme

$$\dot{\hat{X}}^m = A^m(\hat{X}^m, \hat{X}^e) + G_L^m(Z^m - H^m(\hat{X}^m)) + L^m(L^e)^{-1}\dot{\hat{X}}^e,$$

en supposant (L^e) inversible.

Eléments d'analyse L'analyse de cet estimateur se fonde sur une stratégie similaire à celle de l'estimateur état+paramètres (25). Nous la développons ici puisqu'elle n'a pas été publiée. Après linéarisation des dynamiques de l'erreur, on obtient le système de l'erreur linéarisée suivant, initié à partir $\delta\tilde{X}^e(0) = \zeta^e$ et $\delta\tilde{X}^m(0) = \zeta^m$,

$$\begin{cases} \delta\dot{\tilde{X}}^e &= -U^{-1}L^\top (dH)^\top M(dH)\delta\tilde{X}, \\ \delta\dot{\tilde{X}}^m &= (d_m A - G_L^m d_m H^m)\delta\tilde{X}^m + (d_e A)\delta\tilde{X}^e + L^m(L^e)^{-1}\delta\dot{\tilde{X}}^e. \end{cases} \quad (28)$$

On procède alors à un changement de variable comparable à celui introduit dans Moireau et al. (2008)

$$(\delta\tilde{X}^e, \delta\tilde{X}^m) \mapsto (\delta\eta, \delta\tilde{X}^e) = (\delta\tilde{X}^m - L^m(L^e)^{-1}\delta\tilde{X}^e, \delta\tilde{X}^e),$$

et obtenons $\eta(0) = \zeta^m$ (et toujours $\delta\tilde{X}^e(0) = \zeta^e$), ainsi que la dynamique

$$\begin{cases} \delta\dot{\eta} &= (d_m A - G_L^m d_m H^m)\delta\eta, \\ \delta\dot{\tilde{X}}^e &= -U^{-1}L^\top (dH^e)^\top M(dH^e)L\delta\tilde{X}^e U^{-1}L^\top (d_m H^m)^\top M^m(d_m H^m)\delta\eta. \end{cases} \quad (29)$$

La première équation correspond à la dynamique de l'erreur linéarisée pour le système mécanique seul. Nous savons qu'elle converge vers 0 si l'observateur mécanique est efficace. On peut alors démontrer que la seconde équation conduit à une dynamique stabilisée si une condition d'observabilité de la forme

$$\exists(C, T), \quad \int_0^T \|H(L\delta\tilde{X}^e)\|_M^2 \geq C\|\delta\tilde{X}^e(0)\|_{U_\diamond^e}^2,$$

est satisfaite pour toute erreur initiale $\delta\tilde{X}^e(0)$. Cette condition exprime le fait que les variations de la partie électrophysiologique sont perceptibles dans les observations à la fois électriques et mécaniques. On s'attend donc à ce que cette condition soit plus facilement vérifiée que si seules les données électriques permettaient d'estimer le modèle électrophysiologique.

Illustration numérique de l'apport informationnel des données multiples Nous illustrons notre propos par un cas numérique sur des données synthétiques. Le contexte numérique est tiré de Chapelle et al. (2009). L'estimateur utilisé pour ces simulations est une variante UKF de (25) et (27) qui remplace les opérateurs tangents par des interpolations à partir de particules (Moireau et Chapelle, 2011; Chabiniok et al., 2011).

La première illustration numérique concerne le modèle électrique seul. On identifie deux paramètres supposés uniformes dans le myocarde. Les résultats justifient l'utilisation d'un filtre d'état en plus de l'identification paramétrique.

Paramètre	Valeur Cible	<i>A priori</i> (%Erreur)	Identification seule	Estimation conjointe & état+paramètres
τ_{in}	1.0	1.5(50%)	2.19(118.8%)	1.02(1.99%)
τ_{out}	16.0	11.0(31.25%)	6.96(56.48%)	15.15(5.3%)

TAB. 1 – Estimation des paramètres τ_{in} et τ_{out} à partir de mesures ECG

La seconde illustration consiste à estimer un paramètre τ_{close} à différents niveaux de l'épaisseur du ventricule gauche et dans le ventricule droit. On évalue ici l'apport fondamental des mesures de déplacement issues de l'IRM en plus des simples mesures ECG.

Paramètre	Valeur Cible	<i>A priori</i> (Erreur%)	Estimation via ECG	Estimation via ECG+IRM
τ_{close}^{endo}	140.0	56.0 (40%)	121.63 (13.12%)	139.42 (0.41%)
τ_{close}^{mcell}	105.0	42.0 (40%)	70.12 (33.22%)	96.98 (7.64%)
τ_{close}^{epi}	105.0	42.0 (40%)	94.64 (9.87%)	102.68 (2.21%)
τ_{close}^{RV}	120.0	48.0 (40%)	106.94 (10.88%)	116.26 (3.12%)

TAB. 2 – Estimation de τ_{close} avec des mesures ECG seules ou avec des mesures ECG et IRM

Références

- Auroux, D. et J. Blum (2005). Back and forth nudging algorithm for data assimilation problems. *Comptes Rendus Mathématique* 340(12), 873 – 878.
- Bertoglio, C., D. Chapelle, M. Fernández, J.-F. Gerbeau, et P. Moireau (2011). State observers of a vascular fluid-structure interaction model through measurements in the solid. *Comput. Meth. Appl. Mech. Engrg* 256(2013), 149–168.
- Blum, J., F.-X. Le Dimet, et I. M. Navon (2008). Data assimilation for geophysical fluids. In R. Temam et J. Tribbia (Eds.), *Handbook of Numerical Analysis : Computational Methods for the Atmosphere and the Oceans*. Elsevier.
- Boulakia, M., M. Fernández, J.-F. Gerbeau, et N. Zemzemi (2008). Direct and inverse problem in electrocardiography. In P. G. T.E. Simos et C. Tsitouras (Eds.), *Numerical Analysis and Applied Mathematics*, pp. 113–118. American Institute of Physics.

- Boulakia, M., E. Schenone, et J.-F. Gerbeau (2012). Reduced-order modeling for cardiac electrophysiology. application to parameter identification. *International Journal for Numerical Methods in Biomedical Engineering* 28(6-7), 727–744.
- Chabiniok, R., P. Moireau, P.-F. Lesault, A. Rahmouni, J.-F. Deux, et D. Chapelle (2011). Estimation of tissue contractility from cardiac cine-mri using a biomechanical heart model. *Biomechanics and Modeling in Mechanobiology*. published online.
- Chapelle, D., M. Fernández, J.-F. Gerbeau, P. Moireau, J. Sainte-Marie, et N. Zenzemi (2009). Numerical simulation of the electromechanical activity of the heart. In *Proceedings of FIMH'09, LNCS 5528*, pp. 357–365. Springer.
- Chapelle, D., A. Gariah, P. Moireau, et J. Sainte-Marie (2013). A Galerkin strategy with Proper Orthogonal Decomposition for parameter-dependent problems -Analysis, assessments and applications to parameter estimation. *ESAIM : Mathematical Modelling and Numerical Analysis*.
- Chapelle, D., P. Le Tallec, P. Moireau, et M. Sorine (2012). An energy-preserving muscle tissue model : formulation and compatible discretizations. *International Journal for Multiscale Computational Engineering* 10(2), 189–211.
- Julier, S. J. (2002). Reduced Sigma Point Filters for the Propagation of Means and Covariances Through Nonlinear Transformations. In *In Proceedings of the 2002 American Control Conference*, Volume 2, pp. 887–892.
- Moireau, P. et D. Chapelle (2011). Reduced-order Unscented Kalman Filtering with application to parameter identification in large-dimensional systems. *ESAIM : Control, Optimisation and Calculus of Variations*.
- Moireau, P., D. Chapelle, et P. Le Tallec (2008). Joint state and parameter estimation for distributed mechanical systems. *Computer Methods in Applied Mechanics and Engineering* 197, 659–677.
- Moireau, P., D. Chapelle, et P. Le Tallec (2009). Filtering for distributed mechanical systems using position measurements : Perspectives in medical imaging. *Inverse Problems* 25(3), 035010 (25pp). doi :10.1088/0266-5611/25/3/035010.
- Simon, D. (2006). *Optimal State Estimation : Kalman, H^∞ , and Nonlinear Approaches*. Wiley-Interscience.

Summary

This work considers the inverse problem of electrocardiology – namely the inversion of electrocardiogram ECG measurements – from a new perspective. This ill-posed inverse problem is integrated into a more general data assimilation problem combining ECG measurements and medical images of cardiac dynamics. Hence considering the coupled problem of electrophysiology and its cardiac mechanics response, we demonstrate how we can now estimate accurately the state and parameter of the studied system. To do so we rely on a sequential data assimilation strategy which consists of a Luenberger estimator for the cardiac mechanics part and reduced order optimal estimators for the electrical state and the parameters.

Index

Baret, Frédéric, 27

Chabot, Vincent, 1

Corrado, Cesare, 40

Gerbeau, Jean-Frédéric, 40

Hascoët, Laurent, 12

Lauvernet, Claire, 27

Le Dimet, François-Xavier, 27

Moireau, Philippe, 40

Nodet, Maëlle, 1

Papadakis, Nicolas, 1

Vidard, Arthur, 1

EGC 2014

Organisateurs



Sponsors

