



HAL
open science

Uncertainty quantification in data assimilation

Jules Guillot

► **To cite this version:**

Jules Guillot. Uncertainty quantification in data assimilation. Statistics [math.ST]. Université de Bretagne Sud, 2023. English. NNT : 2023LORIS680 . tel-04419979

HAL Id: tel-04419979

<https://theses.hal.science/tel-04419979v1>

Submitted on 26 Jan 2024

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

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

THÈSE DE DOCTORAT DE

L'UNIVERSITÉ BRETAGNE SUD

ÉCOLE DOCTORALE N° 644

Mathématiques et Sciences et Technologies

de l'Information et de la Communication en Bretagne Océane

Spécialité : *Mathématiques et leurs Interactions*

Par

Jules GUILLOT

Quantification des incertitudes en assimilation de données

Thèse présentée et soutenue à Vannes, le 11 décembre 2023

Unité de recherche : LMBA UMR 6205

Thèse N° : 680

Rapporteurs avant soutenance :

Laurent BERTINO Directeur de Recherche, NERSC (Norvège)
Arthur VIDARD Chargé de Recherche, INRIA

Composition du Jury :

Examineurs :	Laurent BERTINO	Directeur de Recherche, NERSC
	Anne CUZOL	Maître de Conférences, LMBA/UBS
	Thomas ROMARY	Maître de Conférences HDR, Mines Paris
	Arthur VIDARD	Chargé de Recherche, INRIA
Dir. de thèse :	Emmanuel FRÉNOT	Professeur, LMBA/UBS
Co-enc. de thèse :	Pierre AILLIOT	Maître de Conférences, LMBA/UBO

Titre : Quantification des incertitudes en assimilation de données

Mot clés : assimilation de données, quantification des incertitudes, EnKF, filtre particulaire, équation de la chaleur, Lorenz-96

Résumé : L'assimilation de données relève du couplage optimal entre un modèle dynamique et des données d'observation permettant de reconstruire l'état caché d'un système, comme en météorologie ou océanographie. Cependant le modèle dynamique et les observations sont généralement imparfaits du fait de la méconnaissance de phénomènes physiques en jeu ou d'erreurs de mesure dans les données par exemple. Quantifier les incertitudes associées au modèle dynamique et aux observations est alors primordial pour que l'assimilation de données permette de retrouver efficacement l'état caché. Cette thèse met en évidence de nouvelles méthodes permettant de quantifier ces incertitudes correspondant aux erreurs de modèle et d'observation. La première méthode présentée permet de modéliser l'erreur de modèle en tenant compte de la physique du système, notamment lorsque la résolution temporelle des observations est faible. La deuxième méthode consiste à augmenter l'état caché avec l'erreur de modèle afin d'estimer conjointement l'état

caché et l'erreur de modèle à l'aide d'un algorithme couramment utilisé en assimilation de données : le filtre de Kalman d'ensemble. La troisième méthode permet d'estimer conjointement l'état caché et les paramètres du terme aléatoire représentant l'erreur de modèle ou d'observation, en combinant le filtre de Kalman d'ensemble avec un autre algorithme usuel en assimilation de données : le filtre particulaire. Cette méthode générique permet aussi d'estimer les paramètres des techniques d'inflation et de localisation utilisées pour diminuer l'impact de l'erreur d'échantillonnage sur le filtre. Les trois méthodes développées dans ce manuscrit sont comparées aux approches traditionnelles pour la quantification des incertitudes en assimilation de données, à travers des expériences numériques basées sur l'équation de la chaleur pour la première méthode, sur une équation d'advection-diffusion pour la deuxième méthode et sur le modèle chaotique de Lorenz-96 pour la troisième méthode.

Title: Uncertainty quantification in data assimilation

Keywords: data assimilation, uncertainty quantification, EnKF, particle filter, heat equation, Lorenz-96

Abstract: Data assimilation corresponds to the optimal coupling between a dynamical model and observation data that allows to retrieve the latent true state of a system, as it is usually done in meteorology or oceanography. However the dynamical model and the observations are generally unperfect due to the lack of knowledge on physical phenomena or measurement errors in the data for instance. Quantifying the uncertainties related to the dynamical model and to the observations is of key importance to efficiently retrieve the true state thanks to data assimilation. This thesis highlights new methods that allow to quantify these uncertainties which correspond to the model and observation errors. The first method allows to model the model error taking into account the physics of the system, notably when the temporal resolution of the observations is low. The second method consists in augmenting the state with the model

error to jointly estimate the true state and the model error thanks to an algorithm currently used in data assimilation: the ensemble Kalman filter. The third method allows to jointly estimate the true state and the parameters of the random term which represents the model or observation error, combining the ensemble Kalman filter with another usual algorithm in data assimilation: the particle filter. This generic method also allows to estimate the parameters of the inflation and localization strategies used to reduce the sampling error impact on the filter. The three methods developed in this manuscript are compared with traditional approaches for uncertainty quantification in data assimilation, through numerical experiences based on the heat equation for the first method, on the advection-diffusion equation for the second method and on the chaotic Lorenz-96 model for the third method.

Remerciements

Je tiens dans un premier temps à remercier mes deux encadrants, Emmanuel Frénod et Pierre Ailliot, qui m'ont donné l'opportunité de réaliser cette thèse. Cette dernière aurait été tout autre sans les conseils avisés d'Emmanuel et l'aide précieuse de Pierre.

Je remercie aussi tous mes co-auteurs pour leur aide et implication dans les différents papiers.

Je remercie les rapporteurs d'avoir pris le temps de lire ce manuscrit ainsi que les membres du jury d'avoir accepté d'assister à ma soutenance.

Je remercie les membres du LMBA de l'UBS pour leur aide, sympathie et les savoirs enseignés il y a quelques années qui m'ont permis d'arriver au doctorat.

Je remercie les différents chercheurs avec qui j'ai pu échanger lors de conférences, pour leurs conseils et intérêt concernant mon sujet de recherche.

Je remercie Pierre pour son amitié et son aide indéfectibles.

Enfin je remercie mes parents et mon frère pour leur amour et encouragement. De même pour la tendresse de mes beaux-parents. Je remercie surtout ma compagne pour son amour et son courage dans les moments parfois difficiles à traverser.

Merci à tous.

Table des matières

1 Introduction générale	8
1.1 Contexte	8
1.2 Algorithmes usuels en assimilation de données séquentielle	10
1.3 Méthodes courantes de quantification des incertitudes en assimilation de données séquentielle	13
1.4 Présentation des travaux de thèse	14
2 Physics Informed Model Error for Data Assimilation	17
2.1 Introduction	17
2.2 The model error in data assimilation	18
2.3 The ensemble Kalman filter	19
2.4 The physics informed model error	20
2.5 Numerical results	22
2.6 Conclusion	30
3 Estimation en ligne de l'erreur de modèle inspirée par la physique	32
3.1 Cas d'étude	34
3.2 Description de l'algorithme utilisé	34
3.3 Résultats numériques	36
3.4 Conclusion	45
4 State and Stochastic Parameters Estimation with Combined Ensemble Kalman and Particle Filters	48
4.1 Introduction	49
4.2 The new PF-EnKF method	51
4.2.1 PF-EnKF for estimating the model or observation error	51
4.2.2 PF-EnKF with inflation and localization	57
4.3 Numerical results	62
4.3.1 Results for the estimation of the model error	63
4.3.2 Results for the PF-EnKF with inflation and localization	67
4.4 Conclusion and perspectives	71
4.5 Appendices	72
4.5.1 PF-EnKF for estimating the observation error	72

4.5.2	RMSE sensitivity to the hyperparameters $\sigma_{\lambda, Q}$ and $\sigma_{l, Q}$ of the PF-EnKF for estimating the model error	76
4.5.3	Sensitivity of the PF-EnKF localization to the number of observed grid points and to the observation error	78
5	Conclusion générale	80
5.1	Bilan	80
5.2	Perspectives	81
A	Équations du filtre de Kalman	83
B	Partial Differential Equations for Oceanic Artificial Intelligence	89
B.1	Introduction	90
B.2	The model	90
B.3	Application to our problem	91
B.3.1	General view of the problem	91
B.3.2	More details on the data	92
B.3.3	Physics behind the model	93
B.3.4	Link with the model	94
B.4	Numerical results	94
B.4.1	Interannual variability	96
B.4.2	2D-extension of the model	96
B.5	Conclusion	100
	Bibliographie	101

Table des figures

2.1	Optimization of σ_{PIME} , σ_{QD} and σ_{QSS} .	24
2.2	Comparison of the evolution of the heat diffusion for the analysis of each algorithm (the more the color is red, the more the temperature is close to zero).	25
2.3	Comparison of the values of X_4^f and X_4^a for the different algorithms.	26
2.4	Temporal evolution of the estimated temperature of the middle point for each method.	27
2.5	Temporal evolution of the estimated temperature of the middle point for each method with $dt = 1.5$.	28
2.6	Global RMSE of each algorithm according to the value of dt .	28
2.7	95% confidence interval for the temporal evolution of the RMSE of PIME.	29
3.1	Images de l'évolution du système obtenue par chaque algorithme jusqu'à $t = 10$ (i.e. $k = 100$).	41
3.2	Images de l'évolution de l'erreur de modèle estimée par chaque algorithme. Pour chaque algorithme, la moyenne empirique des analyses obtenues pour l'erreur de modèle $\eta_k^a = \frac{1}{N} \sum_{i=1}^N \eta_k^{a,i}$ est utilisée à chaque itération k .	42
3.3	Estimation de $v(t)$ et α par les deux algorithmes. Pour chaque paramètre dynamique, l'intervalle de confiance 95% centré en la moyenne est représenté pour chaque algorithme.	43
3.4	Estimation de $X_k[n/2]$ par les deux algorithmes avec leur intervalle de confiance 95% centré en la moyenne.	44
3.5	Estimation de X_{80} par les deux algorithmes en utilisant leur moyenne empirique des analyses X_{80}^a .	44
3.6	Solution stationnaire décrite par l'équation (3.12) avec $\tilde{v} = v(8)$, $\tilde{\alpha} = \alpha$ et $r = 0.005$.	45
4.1	PF-EnKF cycle to estimate Q_t .	55
4.2	PF-EnKF cycle with inflation and localization.	58
4.3	Comparison of the couples of parameters values (each represented by a pixel) using the likelihood (on the top) and the RMSE (on the bottom). The best couple is identified by a red point for each method.	61

4.4	True spatio-temporal evolution of the system (on the left) with the partial and noisy observation data (on the right).	64
4.5	Estimation of the variance parameter $\lambda_{\mathbf{Q},t}$ and spatial scale parameter $l_{\mathbf{Q},t}$ with the PF-EnKF algorithm. For each panel, the 95% confidence interval is plotted using the quantiles of the particle distribution.	65
4.6	Estimation of the spatial scale parameter $l_{\mathbf{Q},t}$ with the PF-EnKF when $p = n$ (the state is fully observed).	66
4.7	Comparison of the estimates of the inflation and localization parameters of the PF-EnKF with the ones of the usual methods. For the PF-EnKF, the inflation parameter $\lambda_{\mathbf{L},t}$ is estimated by $\frac{1}{N_{\theta}} \sum_{j=1}^{N_{\theta}} \lambda_{\mathbf{L},t}^{a,j}$ and the localization parameter $l_{\mathbf{L},t}$ by $\frac{1}{N_{\theta}} \sum_{j=1}^{N_{\theta}} l_{\mathbf{L},t}^{a,j}$.	69
4.8	Comparison of the values of the estimate of the PF-EnKF localization parameter (blue boxplot) with the optimal parameter value obtained by grid search (red point) for different ensemble sizes N_{memb} (4.8a) and for different numbers of observed grid points p when $N_{memb} = 10$ (4.8b).	70
4.9	PF-EnKF cycle to estimate \mathbf{R}_t .	72
4.10	Estimation of the variance parameter $\lambda_{\mathbf{R},t}$ and spatial scale parameter $l_{\mathbf{R},t}$ with the PF-EnKF algorithm (the 95% confidence interval is also plotted for each panel).	75
4.11	RMSE values over time of the PF-EnKF when \mathbf{Q}_t is estimated, for different values of $\sigma_{\lambda,\mathbf{Q}}$ (on the top) and $\sigma_{l,\mathbf{Q}}$ (on the bottom).	77
4.12	Temporal evolution of the estimate of the PF-EnKF localization parameter (blue line) according to the number of observed grid points p in 4.12a and according to the the level of noise in the observation error $\lambda_{\mathbf{R},t}$ (black dashed line) in 4.12b.	79
5.1	Estimation conjointe des paramètres de variance de \mathbf{Q}_t et d'échelle spatiale de \mathbf{R}_t avec le PF-EnKF pour $N_{memb} = N_{\theta} = 100$ (l'intervalle de confiance 95% est tracé pour chaque panel). Ici $\mathbf{Q}_t = \lambda_{\mathbf{Q},t}^2 \mathbf{I}$ avec le paramètre de variance $\lambda_{\mathbf{Q},t} = 1 + 0.5 \sin\left(\frac{t}{10}\right)$ et $\mathbf{R}_t[k, k'] = e^{-\frac{d(k, k')^2}{l_{\mathbf{R},t}^2}}$ pour $(k, k') \in \{1, \dots, p\}^2$ avec le paramètre d'échelle spatiale $l_{\mathbf{R},t} = 0.75 + \mathbf{1}_{[125;250] \cup [375;T]}(t)$.	82
B.1	Localisation of our dataset. It is present in the rectangle 20.	92
B.2	Variations of the spatial average of the temperature over different years. This has been normalized to be compatible with our model.	92
B.3	Sun radiation arriving at the top of the atmosphere for a standard year in $W.m^{-2}$. The variations between different years are very weak and therefore this year is taken as representative.	93
B.4	Comparison between the model outputs (in blue) and the data for an average year (in black).	95
B.5	Input function used for the model of [21].	95

B.6	Comparison between the model outputs (in blue) and the data (in black) for different years.	97
B.7	Grid (B.7a), initial condition (B.7b) and result of the advection model after a few time step (B.7c) using <i>Freefem++</i> with only an advection term and the ODEs.	98
B.8	Finer grid (B.8a), initial condition (B.8b) and result after a few time step (B.8c) for the advection problem and ODEs.	99
B.9	Same situation as in Figure B.8 but with a different time step.	100
B.10	Same situation as in Figure B.9 but with a discontinuous Galerkin basis.	100

Liste des tableaux

2.1	EnKF with the physics informed model error.	22
2.2	Parameters values.	23
2.3	Optimal values for σ_{PIME} , σ_{QD} and σ_{QSS} .	24
2.4	Global RMSE of each algorithm.	25
3.1	Algorithme avec PIME pour l'estimation en ligne de l'erreur de modèle.	36
3.2	Algorithme avec bruit blanc Gaussien pour l'estimation en ligne de l'erreur de modèle.	37
3.3	Valeurs des paramètres communs aux deux algorithmes.	38
3.4	RMSE globale et probabilité de couverture de chaque algorithme pour le paramètre d'advection $v(t)$.	39
3.5	RMSE globale et probabilité de couverture de chaque algorithme pour le paramètre de diffusion α .	39
3.6	RMSE globale et probabilité de couverture de chaque algorithme pour le milieu de la barre.	39
4.1	Comparison between the PF-EnKF with estimated Q_t and the EnKF with true Q_t . Each result corresponds to the average over the 10 experiments \pm the standard deviation.	67
4.2	Comparison between the PF-EnKF with inflation and localization and the combination of the usual inflation and localization methods, when both Q_t and R_t are unknown with $N_{memb} = 10$.	71
B.1	Values of the parameters.	94
B.2	Parameters determined for the different years.	96

Chapitre 1

Introduction générale

1.1 Contexte

L'assimilation de données, initialement utilisée en météorologie et océanographie, consiste à combiner de façon optimale les informations apportées par un modèle dynamique avec celles fournies par des données d'observation, dans le but de reconstruire l'état caché du système étudié. Cet état caché, aussi appelé vrai état, peut par exemple correspondre à la température en un point à un instant donné d'un objet. Le modèle dynamique donne l'évolution a priori de l'état caché, il peut reposer sur de supposés processus physiques en jeu représentés par des équations aux dérivées partielles (EDPs) ou des équations différentielles ordinaires (EDO) comme dans [21]. Le modèle dynamique peut aussi reposer sur des méthodes statistiques utilisant les données d'observation disponibles (par exemple une régression linéaire) comme dans [34]. Les données d'observation utilisées pour l'assimilation peuvent provenir de capteurs *in situ* tels que des bouées mesurant la température de la mer en surface, ou de satellites (*remote sensing*) pour par exemple identifier la position de perturbations nuageuses. En général, le modèle dynamique et les observations sont imparfaits. Le modèle dynamique peut comporter des paramètres inconnus comme dans [21] ou ne pas tenir compte de tous les processus physiques en jeu comme dans [9], il en résulte alors une erreur appelée erreur de modèle. Les observations sont souvent bruitées du fait de l'imperfection de capteurs, générant ainsi une erreur de mesure. Toujours concernant les observations, il y a aussi l'erreur de représentation définie par [31] qui comprend : l'erreur liée à la résolution spatio-temporelle des observations (ne faisant pas apparaître certains processus physiques), l'erreur liée à l'opérateur d'observation qui lie l'état caché à l'observation et l'erreur de prétraitement des données d'observation (utilisation de données non viables). L'ensemble des erreurs de représentation et de mesure correspond à l'erreur d'observation. L'assimilation de données séquentielle est généralement représentée par le modèle espace-état suivant pour $t \in \{1, \dots, T\}$:

$$\begin{cases} \mathbf{x}_t = M(\mathbf{x}_{t-1}) + \boldsymbol{\eta}_t \\ \mathbf{y}_t = H(\mathbf{x}_t) + \boldsymbol{\varepsilon}_t \end{cases}$$

où \mathbf{x}_t est l'état caché dans \mathbb{R}^n et \mathbf{y}_t est l'observation associée dans \mathbb{R}^p . $M(\cdot)$ est le modèle dynamique et $H(\cdot)$ est l'opérateur d'observation. $\boldsymbol{\eta}_t$ et $\boldsymbol{\varepsilon}_t$ sont respectivement les erreurs de modèle et d'observation.

Le filtre de Kalman (voir [32]) est un des algorithmes les plus connus en assimilation de données. Il sert notamment à l'estimation de la trajectoire pour le programme Apollo. Les équations qui le composent sont déterminées facilement dans le cas linéaire Gaussien : c'est-à-dire que le modèle dynamique et l'opérateur d'observation sont linéaires et que les erreurs de modèle et d'observation sont supposées Gaussiennes, voir Annexe A. Le filtre comporte deux étapes : la prédiction et l'analyse. La prédiction consiste à estimer le vrai état à l'instant t à partir des observations précédentes, ce qui revient à étudier la probabilité conditionnelle $p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})$. Ensuite, une fois que l'on bénéficie de la nouvelle observation à l'instant t , la prédiction précédente peut être corrigée : il s'agit de l'étape d'analyse qui se ramène à l'étude de la probabilité conditionnelle $p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t)$. Une troisième étape peut être utilisée, dès que l'on bénéficie de toutes les observations, pour lisser les estimations obtenues au cours du temps, il s'agit de l'étape de réanalyse (ou de lissage) en lien avec la probabilité conditionnelle $p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_T)$. Le filtre de Kalman étendu permet quant à lui de fonctionner aussi lorsque le modèle dynamique et l'opérateur d'observation ne sont pas linéaires.

Les approches ensemblistes basées sur la méthode de Monte Carlo, tel que le filtre de Kalman d'ensemble (EnKF) développé par [17], s'inspirent des équations du filtre de Kalman pour générer à chaque instant un ensemble d'estimateurs de l'état caché appelés membres. La convergence de l'estimation obtenue avec l'EnKF vers l'estimation obtenue avec le filtre de Kalman a été montrée par [33] dans le cas linéaire Gaussien. L'EnKF est particulièrement utilisé en géoscience car il est capable de gérer des systèmes de grandes dimensions. Différentes versions de cette méthode sont présentes dans la littérature (voir [30]). Deux catégories d'EnKFs apparaissent néanmoins : les stochastiques et les déterministes (aussi dits "racines carrées"). Dans la catégorie stochastique les observations sont considérées comme des variables aléatoires Gaussiennes (voir [10]), contrairement aux méthodes déterministes qui ne perturbent pas les observations au sein du filtre pour limiter l'erreur d'échantillonnage comme expliqué par [51].

Un autre type d'approche ensembliste est le filtre particulaire (PF) où les membres générés sont appelés particules. Contrairement à l'EnKF, les observations ne sont pas utilisées pour corriger ces particules mais pour identifier les plus pertinentes en leur attribuant un poids (une probabilité) qui permet d'estimer la probabilité conditionnelle $p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t)$. Les PFs les plus courants sont le *bootstrap* de [25] et le SIR (*sequential importance resampling*) de [40].

pour lequel le ré-échantillonnage des particules en utilisant leur poids est systématiquement réalisé. La dégénérescence du PF est un problème récurrent où le poids d'une particule tend vers 1, faisant converger les autres particules vers cette dernière, cependant d'autres versions du PF permettent de répondre à ce problème (voir [50]). Le PF est capable de gérer des modèles dynamiques non linéaires et ne nécessite pas que les erreurs de modèle et d'observation soient supposées Gaussiennes. Contrairement à l'EnKF, il ne peut aisément gérer des systèmes de grandes dimensions comme montré par [44] : il s'agit de la "malédiction de la dimension".

Une alternative à l'assimilation de données séquentielle est l'assimilation de données variationnelle. Contrairement à l'approche séquentielle où une observation est assimilée à chaque instant, l'approche variationnelle assimile plusieurs observations sur une fenêtre temporelle donnée. On se ramène alors à un problème de minimisation d'une fonction coût, généralement avec une descente de gradient, qui quantifie l'adéquation de l'estimation de l'état caché avec le modèle dynamique et les observations. Cette approche permet de gérer un modèle dynamique non linéaire et ne nécessite pas d'a priori sur la nature statistique des erreurs de modèle et d'observation. Cependant ces dernières sont supposées constantes dans le temps. L'algorithme d'assimilation de données variationnelle le plus courant est celui à 4 dimensions (les 3 dimensions de l'espace et le temps) : le 4DVar. Différentes combinaisons du 4DVar avec l'EnKF ont aussi été élaborées (voir [7]). Les travaux détaillés dans la suite du manuscrit ne traitent pas de l'assimilation de données variationnelle mais séquentielle.

Les filtres couramment utilisés en assimilation de données séquentielle sont détaillés dans la Section [1.2]

1.2 Algorithmes usuels en assimilation de données séquentielle

L'algorithme [1] décrit le fonctionnement du filtre de Kalman lorsque le modèle dynamique et l'opérateur d'observation sont linéaires et lorsque les erreurs de modèle et d'observation sont supposées Gaussiennes. L'initialisation permet de générer une première estimation, avec \mathbf{P}_0^a préalablement définie. Pour ensuite déterminer la prédiction (*forecast*) \mathbf{x}_t^f , le modèle dynamique \mathbf{M} est appliqué à l'analyse précédente \mathbf{x}_{t-1}^a . Dès que la nouvelle observation \mathbf{y}_t est disponible, l'analyse est obtenue en corrigeant la prédiction avec le gain de Kalman \mathbf{K}_t appliqué à l'innovation \mathbf{d}_t . Le gain de Kalman dépend de la matrice de covariance de l'erreur d'observation \mathbf{R} et de la matrice de covariance de la prédiction \mathbf{P}_t^f qui elle-même dépend de la matrice de covariance de l'erreur de modèle \mathbf{Q} . La matrice de covariance de l'analyse \mathbf{P}_t^a , qui dépend du gain de Kalman, servira au calcul de la matrice de covariance de la prédiction à l'itération suivante. Les erreurs de modèle et d'observation ont donc une importance cruciale dans le

fonctionnement du filtre de Kalman.

Algorithm 1 Filtre de Kalman

```

1: Initialisation :
2:  $\mathbf{x}_0^a \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0^a)$ 
3: Prédiction :
4: for  $t = 1, \dots, T$  do
5:    $\mathbf{x}_t^f = \mathbf{M}\mathbf{x}_{t-1}^a$ 
6:    $\mathbf{P}_t^f = \mathbf{M}\mathbf{P}_{t-1}^a\mathbf{M}^\top + \mathbf{Q}$ 
7:    $\mathbf{K}_t = \mathbf{P}_t^f\mathbf{H}^\top (\mathbf{H}\mathbf{P}_t^f\mathbf{H}^\top + \mathbf{R})^{-1}$ 
8:   Analyse :
9:    $\mathbf{d}_t = \mathbf{y}_t - \mathbf{H}\mathbf{x}_t^f$ 
10:   $\mathbf{x}_t^a = \mathbf{x}_t^f + \mathbf{K}_t\mathbf{d}_t$ 
11:   $\mathbf{P}_t^a = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^f$ 
12: end for

```

L'algorithme 2 montre les équations de l'EnKF stochastique (avec perturbation des observations) lorsque l'opérateur d'observation est linéaire et les erreurs de modèle et d'observation sont supposées Gaussiennes. L'initialisation permet de générer un premier ensemble de N membres à propager. Pour obtenir l'ensemble de prédiction à l'instant t , l'ensemble généré à l'analyse à $t - 1$ est propagé par le modèle dynamique puis l'erreur de modèle est rajoutée pour chaque membre. L'ensemble d'analyse est ensuite obtenu en corrigeant l'ensemble de prédiction en utilisant l'observation : chaque prédiction $\mathbf{x}_t^{f,i}$ est corrigée avec le gain de Kalman appliqué à l'innovation liée à la prédiction \mathbf{d}_t^i . Cette innovation dépend de l'erreur de modèle via la prédiction mais aussi de l'erreur d'observation via le terme ϵ_t^i qui perturbe l'observation. Le gain de Kalman est calculé comme dans le filtre de Kalman précédent, où \mathbf{P}_t^f est la matrice de covariance empirique des prédictions. L'EnKF dépend donc particulièrement des erreurs de modèle et d'observation.

Algorithm 2 EnKF stochastique

```
1: Initialisation :
2: for  $i = 1, \dots, N$  do
3:    $\mathbf{x}_0^{a,i} \sim \mathcal{N}(\mathbf{0}, \mathbf{P}_0^a)$ 
4: end for
5: Prédiction :
6: for  $t = 1, \dots, T$  do
7:   for  $i = 1, \dots, N$  do
8:      $\boldsymbol{\eta}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q})$ 
9:      $\mathbf{x}_t^{f,i} = M(\mathbf{x}_{t-1}^{a,i}) + \boldsymbol{\eta}_t^i$ 
10:     $\mathbf{x}_t^f = \frac{1}{N} \sum_{i=1}^N \mathbf{x}_t^{f,i}$ 
11:     $\mathbf{P}_t^f = \frac{1}{N-1} \sum_{i=1}^N (\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)(\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)^\top$ 
12:     $\mathbf{K}_t = \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top + \mathbf{R})^{-1}$ 
13:    Analyse :
14:     $\boldsymbol{\epsilon}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R})$ 
15:     $\mathbf{d}_t^i = \mathbf{y}_t + \boldsymbol{\epsilon}_t^i - \mathbf{H} \mathbf{x}_t^{f,i}$ 
16:     $\mathbf{x}_t^{a,i} = \mathbf{x}_t^{f,i} + \mathbf{K}_t \mathbf{d}_t^i$ 
17:   end for
18: end for
```

L'algorithme [3](#) montre le fonctionnement du PF *bootstrap*. Le modèle dynamique et l'opérateur d'observation peuvent être non linéaires et les erreurs de modèle et d'observation ne sont pas supposées Gaussiennes. Lors de l'initialisation, N particules sont générées suivant une loi initiale choisie. Pour la prédiction, les particules $\mathbf{x}_t^{f,i}$ sont obtenues de la même façon qu'avec l'EnKF, cependant la loi a priori de l'erreur de modèle n'est pas nécessairement une loi normale. Pour l'analyse, l'observation est utilisée pour attribuer un poids γ_t^i à chaque particule de l'ensemble de prédiction. Ce poids est proportionnel à la probabilité que l'erreur d'observation soit égale à l'innovation associée à la particule. Cette probabilité est donnée par la fonction de densité de probabilité f_ϵ de la loi a priori de l'erreur d'observation. Le poids de la particule dépend alors des erreurs d'observation et de modèle. Les N nouvelles particules $\mathbf{x}_t^{a,i}$ obtenues lors de l'analyse suivent la loi mélange $\sum_{j=1}^N \gamma_t^j \delta_{\mathbf{x}_t^{f,j}}$ qui tient compte des prédictions et de leur poids, avec δ la mesure de Dirac. Le PF est donc très dépendant des erreurs de modèle et d'observation.

Algorithm 3 PF *bootstrap*

```
1: Initialisation :
2: for  $i = 1, \dots, N$  do
3:   générer  $\mathbf{x}_0^{a,i}$  suivant la loi initiale
4: end for
5: Prédiction :
6: for  $t = 1, \dots, T$  do
7:   for  $i = 1, \dots, N$  do
8:     générer  $\boldsymbol{\eta}_t^i$  suivant la loi a priori de l'erreur de modèle
9:      $\mathbf{x}_t^{f,i} = M(\mathbf{x}_{t-1}^{a,i}) + \boldsymbol{\eta}_t^i$ 
10:    Analyse :
11:     $\gamma_t^i \propto f_\epsilon(\mathbf{y}_t - H(\mathbf{x}_t^{f,i}))$ 
12:     $\mathbf{x}_t^{a,i} \sim \sum_{j=1}^N \gamma_t^j \delta_{\mathbf{x}_t^{f,j}}$ 
13:   end for
14: end for
```

Les différents algorithmes présentés montrent l'importance des erreurs de modèle et d'observation dans leur fonctionnement. Différentes stratégies pour estimer ces erreurs sont présentées dans la Section [1.3](#).

1.3 Méthodes courantes de quantification des incertitudes en assimilation de données séquentielle.

Dans le filtre, les erreurs de modèle et d'observation sont généralement considérées comme des bruits blancs avec comme matrice de covariance \mathbf{Q} pour l'erreur de modèle et \mathbf{R} pour l'erreur d'observation. L'estimation de \mathbf{Q} et \mathbf{R} est nécessaire pour quantifier les incertitudes liées au modèle dynamique et aux observations, comme illustré par [\[46\]](#). Les méthodes basées sur le maximum de vraisemblance développées hors-ligne (après fonctionnement du filtre) par [\[47\]](#) et en ligne (à chaque itération du filtre) par [\[13\]](#) permettent d'estimer \mathbf{Q} et \mathbf{R} . Le plus souvent ces matrices dépendent de paramètres tels que le niveau de bruit (donnant l'amplitude des valeurs de la matrice) et le paramètre de forme (donnant la structure spatiale de la matrice). Ces paramètres dits stochastiques doivent alors être estimés. Une méthode simple, comme celle utilisée dans [\[48\]](#), consiste à faire tourner le filtre avec différentes valeurs possibles pour ces paramètres, pour ensuite calculer l'erreur d'estimation par rapport à l'état caché associée à chaque valeur de paramètre. Cette méthode est cependant très coûteuse. Différentes combinaisons de l'EnKF avec le PF ont été étudiées par [\[22\]](#) pour estimer les paramètres de \mathbf{R} ou par [\[45\]](#) pour estimer les paramètres de \mathbf{Q} ou \mathbf{R} . Dans ces méthodes l'EnKF génère les membres qui estiment l'état caché tandis que le PF génère les particules qui estiment les paramètres de \mathbf{Q} ou \mathbf{R} .

L'erreur de modèle peut aussi être représentée directement dans le modèle dynamique. Celle-ci peut être estimée par un réseau de neurones comme dans [9] ou représentée par un bruit blanc Gaussien dont la matrice de covariance est à estimer. Ce sont des stratégies utilisées pour la paramétrisation stochastique comme expliqué par [38] et [8]. La méthode développée par [43] permet d'estimer les paramètres de la matrice de covariance du bruit représentant l'erreur de modèle dans le modèle dynamique. Pour cela plusieurs EnKFs sont utilisés, chacun avec des valeurs de paramètres différentes, ce qui rend cette méthode coûteuse.

Estimer conjointement \mathbf{Q} et \mathbf{R} reste une tâche difficile comme expliqué par [46] car ces matrices influencent de la même façon la dispersion des membres générés. Ainsi l'accent peut être mis sur la matrice de covariance des prédictions, plutôt que d'estimer directement \mathbf{Q} et \mathbf{R} . La méthode d'inflation de [6] consiste à multiplier la matrice de covariance des prédictions par un scalaire (représentant le paramètre d'inflation) pour reproduire l'effet de \mathbf{Q} et \mathbf{R} sur cette matrice de covariance. Le paramètre d'inflation est généralement estimé en utilisant les statistiques liées à l'innovation de [16]. Car l'innovation, qui correspond à la différence entre l'observation et la prédiction dans l'espace d'observation, contient des informations sur les erreurs de modèle et d'observation. La méthode d'inflation permet aussi de limiter l'impact de l'erreur d'échantillonnage sur l'estimation du filtre et d'éviter que ce dernier diverge (voir [4]). L'erreur d'échantillonnage est due à une taille d'ensemble générée trop faible à cause de l'utilisation d'un modèle dynamique rendant trop coûteux la propagation d'un nombre important de membres. La méthode de localisation de [29] permet aussi de limiter l'impact de l'erreur d'échantillonnage sur la matrice de covariance des prédictions qui comprend des pseudo-corrélations entre des points éloignés du système. La méthode de localisation supprime ces pseudo-corrélations en appliquant un masque à la matrice de covariance des prédictions, lequel dépend d'un paramètre de localisation à optimiser le plus souvent à la main. D'autres méthodes de localisation sont détaillées dans [19].

Les travaux présentés dans la Section 1.4 traitent de nouvelles approches permettant de quantifier les incertitudes en assimilation de données.

1.4 Présentation des travaux de thèse

Une première approche couplant modèle et données, qui se rapporte plus à la calibration de modèle qu'à l'assimilation de données, est détaillée en Annexe B. Ce travail, publié dans la revue *ESAIM : Proceedings and Surveys*, consiste à appliquer un modèle dynamique utilisé en biologie au problème de l'estimation de la température de la mer en surface. Ce modèle repose sur des EDPs paramétrées représentant des phénomènes d'advection-diffusion en biologie pouvant aussi se retrouver en océanographie. Les paramètres des EDPs sont alors optimisés en utilisant les données de température de la mer en surface dispo-

nibles. La combinaison des informations apportées par le modèle dynamique et les observations a orienté le travail vers les méthodes couramment utilisées en assimilation de données et plus précisément vers la quantification des incertitudes auxquelles ces méthodes doivent faire face. Ainsi, une nouvelle méthode permettant de traiter l'erreur de modèle au sein de l'EnKF est développée dans le Chapitre 2. Ce travail, publié dans la revue *Discrete and Continuous Dynamical Systems-Series S* de l'*American Institute of Mathematical Sciences*, repose sur l'utilisation d'une solution stationnaire bruitée du modèle dynamique pour modéliser l'erreur de modèle. Cette méthode est comparée à des approches couramment utilisées pour modéliser l'erreur de modèle, dans un cas d'étude où le modèle dynamique correspond à l'équation de la chaleur dont la source de chaleur extérieure est inconnue. L'efficacité de cette nouvelle stratégie est plus flagrante lorsque la résolution temporelle des observations est faible. Cette méthode est ensuite utilisée dans le Chapitre 3 pour l'estimation en ligne de l'erreur de modèle avec l'EnKF lorsque l'état est augmenté de l'erreur de modèle et de paramètres dynamiques inconnus. Cette approche est comparée à une méthode équivalente dans un cas d'étude similaire au Chapitre 2 où le modèle dynamique est une équation de la chaleur avec un terme d'advection dont la source de chaleur extérieure et les paramètres de diffusion et d'advection sont inconnus. Cette nouvelle stratégie semble donner une estimation pertinente de l'erreur de modèle et retrouver plus efficacement l'état caché et les paramètres dynamiques que sa méthode concurrente. Cette dernière nécessite aussi l'estimation hors-ligne de paramètres stochastiques qui ne peuvent être estimés en ligne avec l'EnKF. Le PF peut en revanche être utilisé pour l'estimation en ligne de ce type de paramètre : c'est ce qui est proposé dans le Chapitre 4. Ce travail, soumis au journal *Monthly Weather Review* de l'*American Meteorological Society*, apporte une nouvelle combinaison de l'EnKF avec le PF pour estimer conjointement l'état caché et des paramètres stochastiques du schéma d'assimilation. L'EnKF génère des membres estimant l'état caché tandis que le PF génère des particules estimant les paramètres stochastiques. Ces paramètres, variant dans le temps, peuvent être ceux des matrices de covariance des erreurs de modèle \mathbf{Q}_t et d'observation \mathbf{R}_t , ou bien correspondre à des paramètres d'inflation ou de localisation. Cette nouvelle méthode est comparée à des méthodes couramment utilisées pour quantifier les incertitudes liées au modèle dynamique et aux observations et pour réduire l'impact de l'erreur d'échantillonnage sur le filtre. Le modèle dynamique utilisé est le modèle chaotique de Lorenz-96, usuel en assimilation de données. Le nouvel algorithme retrouve correctement l'état caché et donne une estimation pertinente des paramètres stochastiques.

Avant-propos du Chapitre 2

Dans le chapitre qui suit on s'intéresse tout particulièrement à la modélisation de l'erreur de modèle. En suivant le formalisme de [27], on définit ici l'erreur de modèle $\boldsymbol{\eta}_t$ comme étant la différence entre l'état \boldsymbol{x}_t obtenu avec le vrai modèle dynamique et l'état $\hat{\boldsymbol{x}}_t$ obtenu avec un modèle dynamique mal spécifié. On a alors pour chaque instant $t \in \{1, \dots, T\}$

$$\boldsymbol{\eta}_t = \boldsymbol{x}_t - \hat{\boldsymbol{x}}_t.$$

On se place dans le cas où le vrai modèle dynamique est une EDP linéaire inhomogène (avec second membre) et le modèle dynamique mal spécifié correspond à la même EDP mais homogène (sans second membre). On peut alors vérifier que l'erreur de modèle $\boldsymbol{\eta}_t$ est aussi solution du vrai modèle dynamique. De plus, lorsque le pas de temps entre deux observations consécutives est grand par rapport à la vitesse de convergence vers le régime stationnaire du système, il semble alors naturel de modéliser l'erreur de modèle en utilisant la solution stationnaire du vrai modèle dynamique perturbée par un terme aléatoire. Cette approche est détaillée ci-après dans le cas de l'équation de la chaleur avec second membre inconnu.

Chapitre 2

Physics Informed Model Error for Data Assimilation

Jules Guillot, Emmanuel Frénod and Pierre Ailliot

Abstract

Data assimilation consists in combining a dynamical model with noisy observations to estimate the latent true state of a system. The dynamical model is generally misspecified and this generates a model error which is usually treated using a random noise. The aim of this paper is to suggest a new treatment for the model error that further takes into account the physics of the system : the physics informed model error. This model error treatment is a noisy stationary solution of the true dynamical model. It is embedded in the ensemble Kalman filter (EnKF), which is a usual method for data assimilation. The proposed strategy is then applied to study the heat diffusion in a bar when the external heat source is unknown. It is compared to usual methods to quantify the model error. The numerical results show that our method is more accurate, in particular when the observations are available at a low temporal resolution.

2.1 Introduction

This paper is part of the framework of data assimilation. In meteorology and oceanography, see [24], or for factory issues as explained in [1], the goal of data assimilation is to reconstruct the real state of a system, for instance the sea surface temperature, taking into account observation data. It consists in combining a dynamical model, which describes the temporal evolution of the system, with observation data that often contain measurement errors due to the sensors. The EnKF developed in [17] is a usual data assimilation method. The principle of this algorithm is to compute an ensemble of possible trajectories for the system, called the ensemble members, which are then corrected using

the observation data. However the dynamical model embedded in this method generates an error, with respect to the reality (truth), complicated to quantify. This model error is usually treated using a stochastic term with zero mean and a variance to specify. The main contribution of the paper is to propose an original treatment for the model error which further takes into account the physics of the system. This novel method is applied here to a simple case study : the heat equation. The model error is treated by using a particular solution of this equation that includes a stochastic part. The efficiency of this method is then assessed by comparing its error to the ones of usual methods.

In Section [2.2](#) the model error is defined and different methodologies to mitigate it are given. The EnKF is then detailed in Section [2.3](#). The Section [2.4](#) describes the case study and the new model error treatment. Numerical results are then discussed in Section [2.5](#). Finally, some remarks and perspectives are given in Section [2.6](#).

2.2 The model error in data assimilation

Two steps are involved in the sequential data assimilation process : the forecast and the analysis. For the forecast, the dynamical model is often used with an additive Gaussian white noise, to compensate the lack of knowledge due to unknown physical phenomena, or to correct a wrong parameterization of the dynamical model. The latter can be determined by two different approaches. If physical processes of the system are known, parameterized partial differential equations (PDEs) or ordinary differential equations (ODEs) are solved : this is the model-driven approach. Whereas when the underlying phenomena are unknown, a machine learning tool based on the available observation data may be used : this is the data-driven approach, as illustrated in [34](#). For the analysis, the new observation is used to modify the forecast. The real state is in general partially observed by the sensors, so the observation corresponds to the observed parts of the real state. The observation is estimated by applying an observation operator to the forecast. This observation operator returns the forecast for the observed parts of the real state. The difference between the observation and its estimate is called the innovation. The Kalman gain is then applied to this innovation to correct the forecast. It is the optimal operator in the sense that when applied to the innovation it is supposed to minimize the mean square error with respect to the real state.

The additive noise in the forecast step is used to probe the difference between the real state and the output of the dynamical model, which is the solution of the dynamical model applied to the previous state estimate. This difference is the consequence of the model error. Estimating the covariance matrix of the model error treatment allows to quantify the model uncertainty, different techniques based on the method of moments or the maximum likelihood of the innovation are shown in [46](#).

The inflation method established by [6](#) is usually used to correct the bad forecasting especially due to the treatment of the model error. It consists in

increasing the coefficients of the covariance matrix of the forecasts, multiplying this matrix by a time-dependent scalar determined thanks to the equations of [16]. A temporal smoothing could be necessary if this inflation factor is too noisy.

Another approach is detailed in [9] and [20], where the model error is treated using the output of a neural network.

Stochastic parametrization is also frequently used for numerical weather prediction. Instead of adding a treatment for the model error to the output of the dynamical model, the unknown physical process is directly replaced inside the dynamical model by a stochastic term. The PDEs involved in the dynamical model become Stochastic-PDEs (SPDEs) but are solved with high computational cost to obtain the forecasts. Different strategies of stochastic parametrizations are described in [38] and [8].

The new approach detailed in this paper is slightly different from the others : the model error is treated using a randomized solution of the stationary PDE involved in the dynamical model.

2.3 The ensemble Kalman filter

Sequential data assimilation is based on the following state-space model

$$\begin{cases} X_k = M(X_{k-1}) + \eta_k & (2.1a) \\ Y_k = H(X_k) + \varepsilon_k & (2.1b) \end{cases}$$

where $X_k \in \mathbb{R}^n$ is the real state at the discrete time $t_k = (k-1)dt$, with $k \in \{1, \dots, K_{final}\}$ and dt is the time step. $Y_k \in \mathbb{R}^p$ is the related observation. M is the dynamical model applied to the previous state and H is the observation operator giving the observed components of X_k . $\eta_k \sim \mathcal{N}(0, Q)$ and $\varepsilon_k \sim \mathcal{N}(0, R)$ are the uncertainties with Gaussian assumptions, respectively the treatments for the model error and the observation error. Indeed the observation contains an error, due to the sensor, which is often treated using a Gaussian white noise with a covariance matrix R to specify, but this is not the purpose of the paper. The goal is to estimate the real state given the dynamical model and the observation data.

The EnKF provides an ensemble of N estimates for the real state at each time t_k . During the forecast step, the EnKF determines the N forecasts $X_k^{f,i}$ for $i \in \{1, \dots, N\}$, thanks to the dynamical model and the treatment for the model error. Then knowing the new observation Y_k , those forecasts are corrected using the innovation d_k^i to obtain the N analyses $X_k^{a,i}$. $X_1^{a,i}$ corresponds to the initial condition X_0 of the dynamical model, plus $\eta_1^i \sim \mathcal{N}(0, Q)$, to generate the ensemble members for the first iteration.

Forecast step :

$$X_k^{f,i} = M(X_{k-1}^{a,i}) + \eta_k^i \quad \eta_k^i \sim \mathcal{N}(0, Q) \quad (2.2)$$

$$X_k^f = \frac{1}{N} \sum_{i=1}^N X_k^{f,i} \quad (2.3)$$

$$P_k^f = \frac{1}{N-1} \sum_{i=1}^N (X_k^{f,i} - X_k^f)(X_k^{f,i} - X_k^f)^\top \quad (2.4)$$

P_k^f is the estimated covariance matrix of the forecasts and notably depends on Q .

Analysis step :

$$K_k = P_k^f H^\top (H P_k^f H^\top + R)^{-1} \quad (2.5)$$

$$d_k^i = Y_k + \varepsilon_k^i - H X_k^{f,i} \quad \varepsilon_k^i \sim \mathcal{N}(0, R) \quad (2.6)$$

$$X_k^{a,i} = X_k^{f,i} + K_k d_k^i \quad (2.7)$$

$$X_k^a = \frac{1}{N} \sum_{i=1}^N X_k^{a,i} \quad (2.8)$$

The estimation of Q and R is a key point in data assimilation as explained in [46]. Most of the time, the model error treatments are uncorrelated in space. A strategy to further take into account the physics and mitigate the model error is discussed hereafter.

2.4 The physics informed model error

The new approach for the treatment of the model error is applied to the assimilation of temperature data of a metallic bar heated on its center. The real state is the solution of the following inhomogeneous heat equation

$$\frac{\partial X}{\partial t}(x, t) - \alpha \frac{\partial^2 X}{\partial x^2}(x, t) = r(t) \quad \text{for } x \in [0, 1] \text{ and } t > 0 \quad (2.9)$$

with the initial and boundary conditions

$$X(x, 0) = \sin(\pi x) \quad \text{and} \quad X(0, t) = X(1, t) = 0. \quad (2.10)$$

$X(x, t)$ is the real temperature at the position x on the bar at time t . The goal is to estimate $X(x, t)$ without knowing the external heat source $r(t)$ on the

right-hand side of equation (2.9), using the EnKF with the following dynamical model and model error treatment.

The dynamical model :

The misspecified dynamical model used is the homogeneous heat equation :

$$\frac{\partial X}{\partial t}(x, t) - \alpha \frac{\partial^2 X}{\partial x^2}(x, t) = 0 \quad (2.11)$$

with the same initial and boundary conditions as (2.10).

The space is then discretized such as $0 = x_1 < \dots < x_j < \dots < x_n = 1$, with dx the space step. The semi-discretization in space of (2.11) is led by using the centered finite difference scheme

$$\frac{\partial X}{\partial t}(x_j, t) = \alpha \frac{X(x_{j+1}, t) - 2X(x_j, t) + X(x_{j-1}, t))}{(dx)^2} \quad \text{for } j \in [2, \dots, n-1] \quad (2.12)$$

with $X(x_j, 0) = \sin(\pi x_j)$ and $X(x_1, t) = X(x_n, t) = 0$.

Then this ODE is discretized in time and solved using a RK4-5 scheme.

The model error treatment (physics informed model error) :

As the external heat source $r(t)$ is unknown, the use of a relevant treatment for the model error is necessary to compensate this lack of knowledge. The solution of the stationary heat equation with noisy right-hand side is chosen to treat the model error. To determine it we solve the related stationary equation with $r(t) = r$ unknown constant

$$\begin{aligned} -\alpha \frac{\partial^2 X}{\partial x^2}(x, t) &= r && \text{since } \frac{\partial X}{\partial t}(x, t) = 0 \\ \iff \frac{\partial X}{\partial x}(x, t) &= -\frac{r}{\alpha}x + c_1 \\ \iff X(x, t) &= -\frac{r}{2\alpha}x^2 + c_1x + c_2. \end{aligned}$$

The constants c_1 and c_2 are determined thanks to the boundary conditions

$$\begin{aligned} X(0, t) &= c_2 = 0 \\ X(1, t) &= -\frac{r}{2\alpha} + c_1 = 0 \Rightarrow c_1 = \frac{r}{2\alpha}. \end{aligned}$$

So the solution of the stationary equation is $w(x) = \frac{r}{2\alpha}(-x^2 + x)$. This solution is made noisy by replacing r by a one-dimensional noise $r_k^i \sim \mathcal{N}(0, \sigma^2)$, which refers to the ensemble member i and the time t_k for the forecast step. The parameter σ is to specify by practice.

TABLE 2.1 – EnKF with the physics informed model error.

<p><i>Initialization</i> : for $i = 1, \dots, N$ generate w_1^i $X_1^{a,i} = X_0 + w_1^i$ For $k \geq 2$ <i>Forecast</i> : for $i = 1, \dots, N$ generate w_k^i $X_k^{f,i} = M(X_{k-1}^{a,i}) + w_k^i$ $X_k^f = \frac{1}{N} \sum_{i=1}^N X_k^{f,i}$ $P_k^f = \frac{1}{N-1} \sum_{i=1}^N (X_k^{f,i} - X_k^f)(X_k^{f,i} - X_k^f)^\top$ <i>Analysis</i> : for $i = 1, \dots, N$ generate $\varepsilon_k^i \sim \mathcal{N}(0, R)$ $K_k = P_k^f H^\top (H P_k^f H^\top + R)^{-1}$ $d_k^i = Y_k + \varepsilon_k^i - H X_k^{f,i}$ $X_k^{a,i} = X_k^{f,i} + K_k d_k^i$ $X_k^a = \frac{1}{N} \sum_{i=1}^N X_k^{a,i}$</p>

The resulting model error treatment is

$$w_k^i = \begin{pmatrix} w_k^i(x_1) \\ \vdots \\ w_k^i(x_j) \\ \vdots \\ w_k^i(x_n) \end{pmatrix} \quad \text{with } w_k^i(x_j) = \frac{r_k^i}{2\alpha} (-x_j^2 + x_j). \quad (2.13)$$

The algorithm used to estimate the temperature $X(x, t)$ with the new model error treatment is detailed in Table 2.1. In the forecast step, the term $M(X_{k-1}^{a,i})$ corresponds to the solution at $t = dt$ of the dynamical model with the initial condition X_0 replaced by $X_{k-1}^{a,i}$. In the analysis step, p points on the bar among n are observed, these observation data are contained in Y_k .

2.5 Numerical results

The efficiency of the treatment of the model error previously detailed is assessed by comparing this method to reference methods used to mitigate the model error in the EnKF. The following algorithms used for the comparison embed the same dynamical model for the forecast step, which corresponds to the homogeneous heat equation (2.11) where the diffusion coefficient α is known. However the treatment for the model error in the forecast step is different for

each algorithm. The analysis step is the same for all the algorithms, with the same observations simulated following (2.1b).

The algorithm that treats the model error using the method in Section 2.4 is called the Physics Informed Model Error algorithm (PIME) and the associated parameter σ is denoted as σ_{PIME} . The other algorithms used for the comparison treat the model error using a Gaussian white noise η_k^i , see (2.2), but with a different covariance matrix Q . One uses $Q = \sigma_{QD}^2 I_n$ (I_n is the n by n identity matrix) : it is called the Q -Diagonal algorithm (QD). Whereas the other uses $Q[i, j] = \sigma_{QSS}^2 e^{-\lambda|x_i - x_j|}$ for $1 \leq i, j \leq n$: it is called the Q -Spatial Structure algorithm (QSS) because it takes into account the position on the bar. The values of the diagonal of this covariance matrix are equal to σ_{QSS}^2 and the more the elements of the matrix are away from the diagonal, the more the values of these elements are close to zero, with a speed that depends on the value of the spatial scale parameter λ . The parameters σ_{PIME} , σ_{QD} , σ_{QSS} and λ are to optimize by practice.

As said in Section 2.4 the real state is the solution of equation (2.9). The goal of this study is to compare the efficiency of each algorithm to retrieve the real state, without knowing the external heat source $r(t)$.

For the real state we put $r(t) = 0.1 \sin(t)$, to have an external heat source that will alternatively reheat or refresh the bar over time.

The parameters values for the three algorithms are given in Table 2.2. Among the n points of interest on the bar, only one in two is observed with low temporal frequency (every time period dt) and with an important observation noise (depending on R).

Parameter	Value
n	100
p	50
N	30
R	$0.01I_p$
K_{final}	30
α	0.05
dt	1

TABLE 2.2 – Parameters values.

Each algorithm is assessed using the global Root Mean Square Error

$$\text{Global RMSE} = \frac{1}{K_{final}} \sum_{k=1}^{K_{final}} \sqrt{\frac{1}{nN} \sum_{i=1}^N (X_k^{a,i} - X_k)^\top (X_k^{a,i} - X_k)}. \quad (2.14)$$

For QSS, the global RMSE is computed for different values of σ_{QSS} and λ . The global RMSE of QSS is more sensitive to the value of σ_{QSS} than to the value of λ . To minimize the global RMSE of QSS with respect to λ , we put

$\lambda = 0.01$.

The parameters σ_{PIME} , σ_{QD} and σ_{QSS} are optimized by minimizing the global RMSE for each algorithm. To this end, we put $\sigma_{PIME} = \sigma_{QD} = \sigma_{QSS} = \sigma$ and vary σ using the logarithmic scale $\{10^{-5}, 10^{-4.9}, 10^{-4.8}, \dots, 10^{-0.1}, 1\}$. The global RMSE of each algorithm for each value of σ is plotted in Figure 2.1

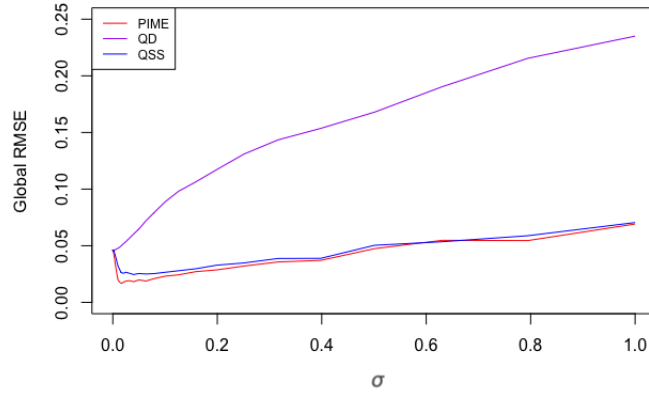


FIGURE 2.1 – Optimization of σ_{PIME} , σ_{QD} and σ_{QSS} .

For each algorithm, the value of σ that minimizes the global RMSE is shown in Table 2.3. The numerical results reported hereafter are obtained using these values.

σ_{PIME}	0.016
σ_{QD}	0.001
σ_{QSS}	0.050

TABLE 2.3 – Optimal values for σ_{PIME} , σ_{QD} and σ_{QSS}

The temporal evolution of the heat diffusion in the bar is shown in Figure 2.2. The average of the analysis ensemble X_k^a is used to obtain the image of the diffusion for each algorithm. Contrary to QD and QSS, PIME partially retrieves the reheating of the bar (at t_9 , t_{15} and t_{28}) due to the unknown external heat source.

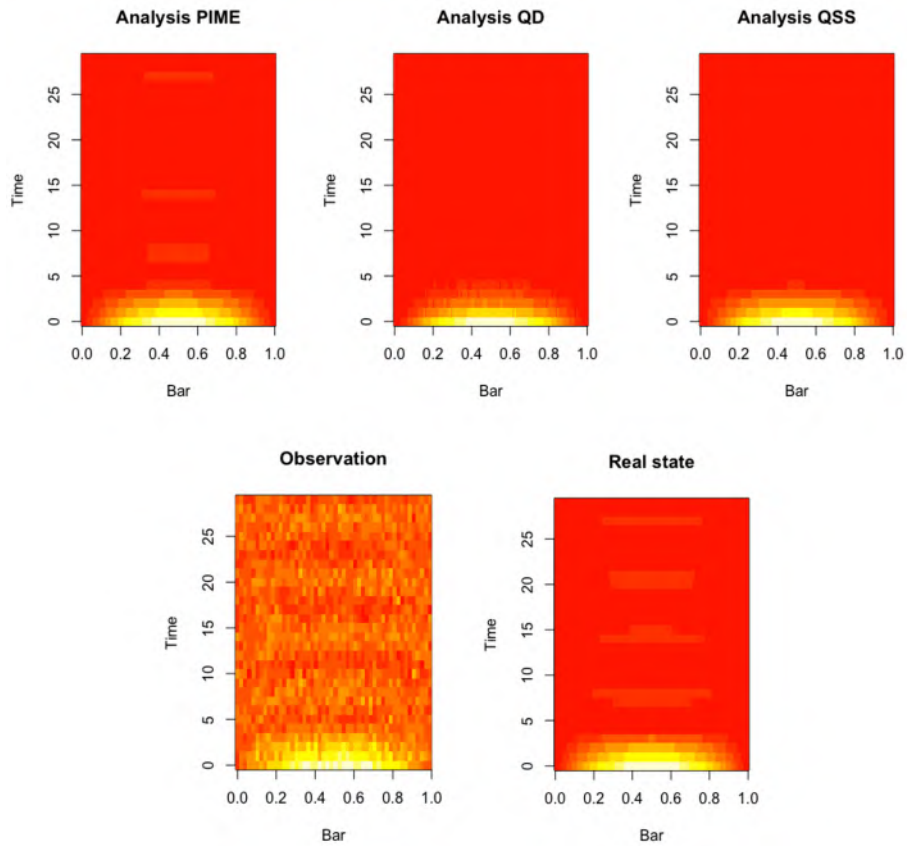


FIGURE 2.2 – Comparison of the evolution of the heat diffusion for the analysis of each algorithm (the more the color is red, the more the temperature is close to zero).

The global RMSE is computed for each algorithm and given in Table 2.4. PIME is the most accurate, followed by QSS and QD.

Algorithm	Global RMSE
PIME	0.017
QD	0.048
QSS	0.025

TABLE 2.4 – Global RMSE of each algorithm.

Focusing on time t_4 , X_4^f and X_4^a are plotted for each algorithm with the real state X_4 and the observation Y_4 in Figure 2.3

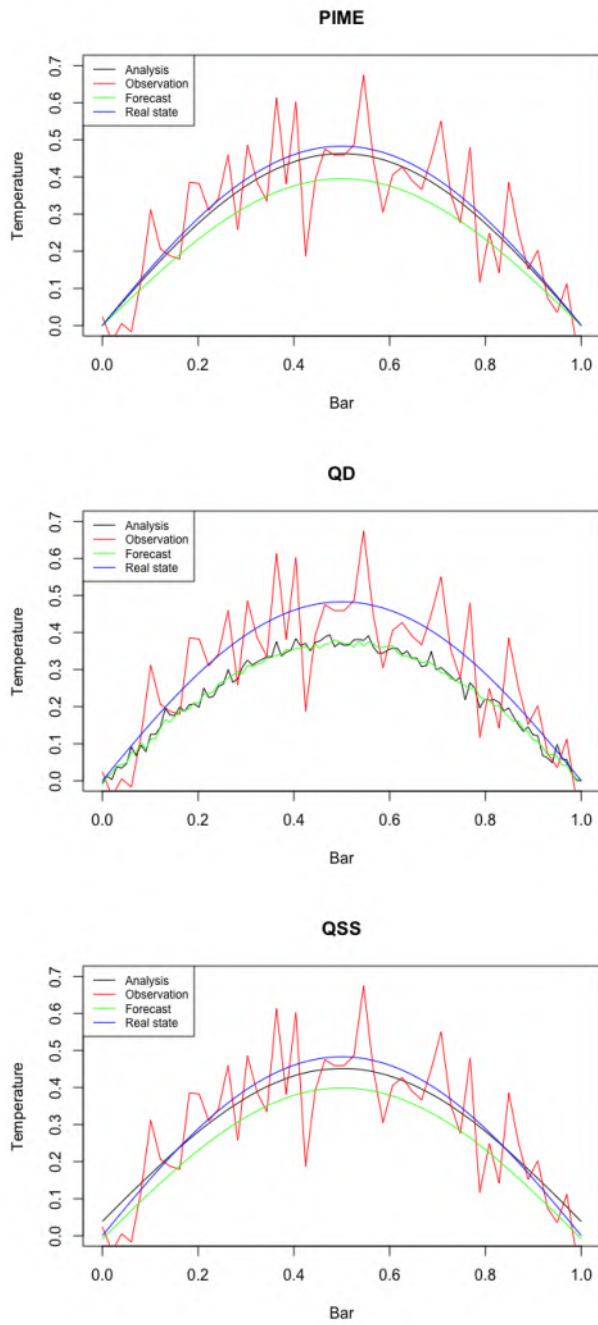


FIGURE 2.3 – Comparison of the values of X_4^f and X_4^a for the different algorithms.

In Figure 2.3 for the three algorithms the analysis is better than the forecast as expected. PIME gives the best analysis. The one of QSS is slightly less accurate, mainly at the boundaries of the bar. QD gives the worst analysis and the temperature fields are not smooth in space : this is a consequence of using a diagonal matrix for Q and thus no spatial correlation for the model error treatment.

The efficiency of each algorithm is now studied for the middle of the bar, which is the point where the error is the most important. To achieve this, the value of the 50th component of X_k^a is plotted as a function of t_k for each algorithm. In Figure 2.4 PIME and QSS can retrieve the oscillations of the temperature due to $r(t)$, with a better accuracy for PIME. Whereas QD is unable to reconstruct these oscillations.

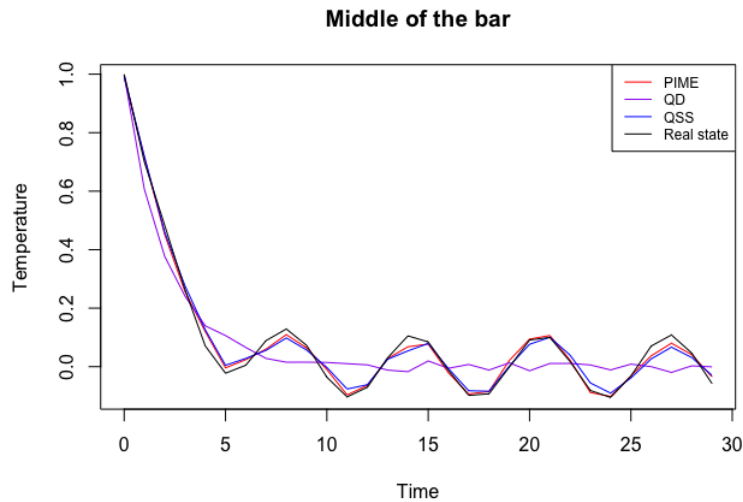


FIGURE 2.4 – Temporal evolution of the estimated temperature of the middle point for each method.

The same study is led when the time period between two consecutive observations is increased to $dt = 1.5$. As there are less observations that contain informations about $r(t)$, the model error is more important and its treatments are to adapt by optimizing σ_{PIME} , σ_{QD} and σ_{QSS} with the method related to Figure 2.1 PIME again gives the best estimate in Figure 2.5 and widens the gap with the other methods.

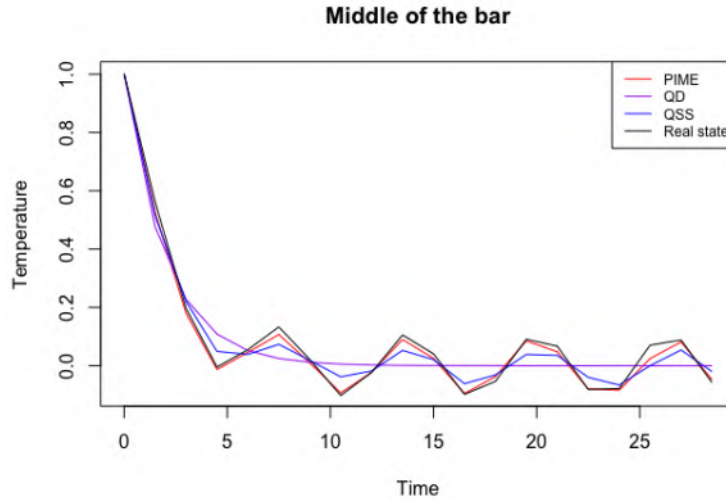


FIGURE 2.5 – Temporal evolution of the estimated temperature of the middle point for each method with $dt = 1.5$.

The efficiency of PIME, in comparison with the other algorithms, is more obvious when dt is important. This is also shown in Figure 2.6, where the global RMSE for each value of dt is plotted for each algorithm. The parameters σ_{PIME} , σ_{QD} and σ_{QSS} are optimized for each value of dt .

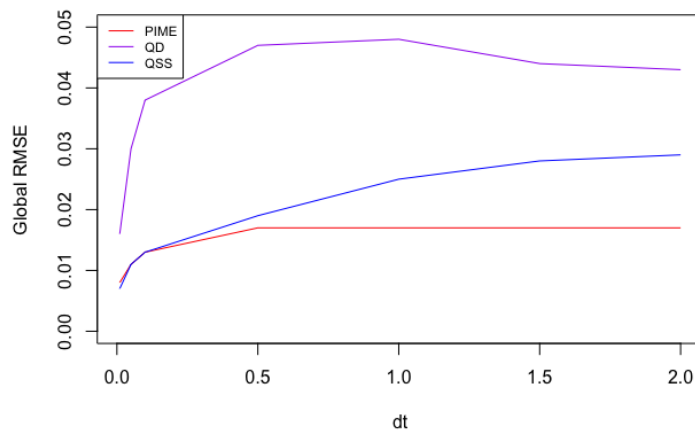


FIGURE 2.6 – Global RMSE of each algorithm according to the value of dt .

When dt is smaller, PIME and QSS are more accurate and have nearly the same error. There are more observations over time that give informations on the

real states, that is why the error of each algorithm is less important. The added value of PIME cannot be seen easily when dt is too small, because the system does not evolve enough between two consecutive times to let appear a sort of stationary state which is approached by PIME thanks to its physics informed model error.

Lastly, to check if the results discussed above for PIME are robust versus random perturbations in the observation noise and in the noise of the model error treatment, a 95% confidence interval for the temporal evolution of the RMSE of PIME is built. To this end, the experience of the heat diffusion in the bar is repeated 100 times for PIME with the parameters values of Table 2.2 and $\sigma_{PIME} = 0.016$. The real state, dynamical model and model error remain the same, whereas the values of the observations and of the model error treatment vary from one experience to another through their noises. At each experience, the temporal evolution of the RMSE of PIME is determined computing the RMSE for each time t_k

$$\text{RMSE}(t_k) = \sqrt{\frac{1}{nN} \sum_{i=1}^N (X_k^{a,i} - X_k)^\top (X_k^{a,i} - X_k)}. \quad (2.15)$$

The average of these temporal evolutions of RMSE is plotted in red in Figure 2.7 and the 95% confidence interval in grey is computed for each time t_k .

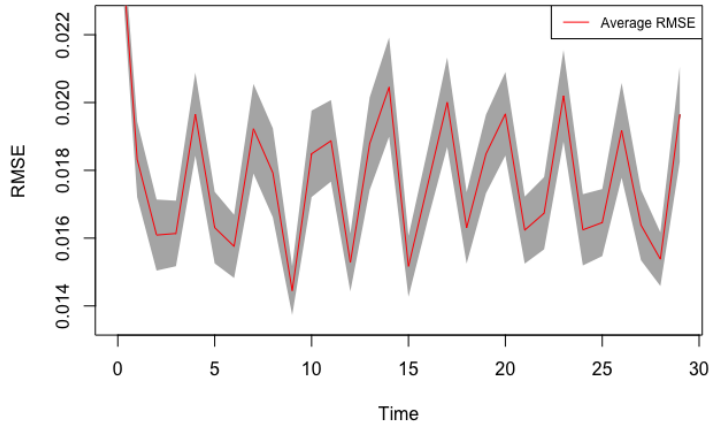


FIGURE 2.7 – 95% confidence interval for the temporal evolution of the RMSE of PIME.

The evolution of the RMSE depends on the evolution of the external heat source : larger RMSE values are obtained when $|r(t_k)|$ is larger. The peak at t_1 is due to $X_1^{a,i}$ for which the initial condition X_0 is perturbed by w_1^i to generate the ensemble members (see Table 2.1), producing first a relatively important

RMSE that then decreases using the dynamical model in the forecast step at t_2 . This peak is also observed for the other algorithms and for the same reason. The confidence interval is globally narrow. This shows that PIME is stable with respect to the observation noise and the noise in the model error treatment.

2.6 Conclusion

A new treatment for the model error in the EnKF was introduced, taking further into account the physics of the system. This treatment is a randomized solution of the stationary PDE associated to the dynamical model. This method was applied to the heat diffusion in a bar, where the goal was to estimate the temperature of the bar over time without knowing the external heat source. In this context, our method was compared to reference methods to mitigate the model error. The numerical results showed the efficiency of our physics informed model error, especially when the frequency of the observations was low.

The online adaption of our method may be interesting to further vary in time the physics informed model error and could lead to better results.

The physics informed model error may be extended to more complex physical phenomena as for the estimation of the sea surface temperature.

Contributions :

- Nouvelle approche pour modéliser l'erreur de modèle tenant compte de la corrélation spatiale des points et des paramètres dynamiques,
- Cette méthode permet de réduire significativement l'erreur de modèle lorsque la résolution temporelle des observations est faible.

Avant-propos du Chapitre 3

Dans le Chapitre 2 la modélisation de l'erreur de modèle n'est pas structurée dans le temps puisque le bruit utilisé dans la solution stationnaire est simulé de manière indépendante d'une itération à l'autre. On propose maintenant d'introduire une structure temporelle dans la modélisation de l'erreur de modèle. En pratique, l'état est augmenté avec l'erreur de modèle, qui est modélisée par une marche aléatoire, ce qui permet alors une estimation en ligne de l'erreur de modèle avec l'EnKF. Cette méthode est détaillée dans le cas d'une équation d'advection-diffusion dont le second membre et les paramètres dynamiques sont inconnus.

Chapitre 3

Estimation en ligne de l'erreur de modèle inspirée par la physique

L'estimation conjointe de l'état avec l'erreur de modèle peut être réalisée en augmentant l'état avec l'erreur de modèle en suivant la technique du *state augmentation* généralement utilisée pour estimer des paramètres dynamiques en ligne avec l'EnKF (voir [28]). Une approche légèrement différente a été réalisée par [54] où l'erreur de modèle dépend d'un biais constant en temps qui est estimé par *state augmentation*. Dans ce chapitre, on considère que l'erreur de modèle résulte du fait que le modèle dynamique ne tient pas compte de tous les phénomènes physiques en jeu et que des paramètres dynamiques sont inconnus. On s'intéresse au modèle espace-état suivant pour $k \in \{1, \dots, K_{final}\}$

$$\begin{cases} X_k = M(X_{k-1}; \theta_{k-1}) + \eta_{k-1} & (3.1a) \\ \eta_k = \eta_{k-1} + \tilde{\eta}_k(\theta_{k-1}) & (3.1b) \\ \theta_k = \theta_{k-1} + \tilde{\theta}_k & (3.1c) \\ Y_k = HX_k + \varepsilon_k & (3.1d) \end{cases}$$

avec l'état $X_k \in \mathbb{R}^n$, l'erreur de modèle $\eta_k \in \mathbb{R}^n$, le vecteur de paramètres dynamiques $\theta_k \in \mathbb{R}^{n_\theta}$ et l'observation $Y_k \in \mathbb{R}^p$. $M(\cdot; \theta_{k-1})$ est le modèle dynamique utilisé avec les paramètres dynamiques de θ_{k-1} . L'erreur de modèle est modélisée par une marche aléatoire avec $\tilde{\eta}_k(\theta_{k-1})$ un terme aléatoire qui dépend des paramètres dynamiques car l'erreur de modèle résulte notamment de la méconnaissance des paramètres dynamiques. $\{\theta_k\}_{k \in \{1, \dots, K_{final}\}}$ est une marche aléatoire avec $\tilde{\theta}_k \sim \mathcal{N}(0, C)$. L'opérateur d'observation H est linéaire et l'erreur d'observation est représentée par $\varepsilon_k \sim \mathcal{N}(0, R)$.

L'état augmenté avec l'erreur de modèle et les paramètres dynamiques

$\widehat{X}_k = (X_k \quad \eta_k \quad \theta_k)^\top$ est estimé avec l'EnKF suivant les équations pour $i \in \{1, \dots, N\}$

$$\widehat{X}_k^{f,i} = \begin{pmatrix} X_k^{f,i} \\ \eta_k^{f,i} \\ \theta_k^{f,i} \end{pmatrix} = \begin{pmatrix} M(X_{k-1}^{a,i}; \theta_{k-1}^{a,i}) + \eta_{k-1}^{a,i} \\ \eta_{k-1}^{a,i} + \tilde{\eta}_k^i(\theta_{k-1}^{a,i}) \\ \theta_{k-1}^{a,i} + \tilde{\theta}_k^i \end{pmatrix} \text{ avec } \tilde{\theta}_k^i \sim \mathcal{N}(0, C) \quad (3.2)$$

$$\widehat{X}_k^f = \frac{1}{N} \sum_{i=1}^N \widehat{X}_k^{f,i} \quad (3.3)$$

$$\widehat{P}_k^f = \frac{1}{N-1} \sum_{i=1}^N (\widehat{X}_k^{f,i} - \widehat{X}_k^f)(\widehat{X}_k^{f,i} - \widehat{X}_k^f)^\top = \begin{pmatrix} \text{Var}(X_k^{f,i}) & \text{Cov}(X_k^{f,i}, \eta_k^{f,i}) & \text{Cov}(X_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\eta_k^{f,i}, X_k^{f,i}) & \text{Var}(\eta_k^{f,i}) & \text{Cov}(\eta_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\theta_k^{f,i}, X_k^{f,i}) & \text{Cov}(\theta_k^{f,i}, \eta_k^{f,i}) & \text{Var}(\theta_k^{f,i}) \end{pmatrix} \quad (3.4)$$

$$\begin{aligned} \widehat{K}_k &= \widehat{P}_k^f \widehat{H}^\top (\widehat{H} \widehat{P}_k^f \widehat{H}^\top + R)^{-1} \text{ avec } \widehat{H} = \begin{pmatrix} H & 0_{p \times (n+n_\theta)} \end{pmatrix} \\ &= \begin{pmatrix} \text{Var}(X_k^{f,i}) & \text{Cov}(X_k^{f,i}, \eta_k^{f,i}) & \text{Cov}(X_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\eta_k^{f,i}, X_k^{f,i}) & \text{Var}(\eta_k^{f,i}) & \text{Cov}(\eta_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\theta_k^{f,i}, X_k^{f,i}) & \text{Cov}(\theta_k^{f,i}, \eta_k^{f,i}) & \text{Var}(\theta_k^{f,i}) \end{pmatrix} \begin{pmatrix} H^\top \\ 0_{(n+n_\theta) \times p} \end{pmatrix} \\ &= \begin{pmatrix} (H \quad 0_{p \times (n+n_\theta)}) \begin{pmatrix} \text{Var}(X_k^{f,i}) & \text{Cov}(X_k^{f,i}, \eta_k^{f,i}) & \text{Cov}(X_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\eta_k^{f,i}, X_k^{f,i}) & \text{Var}(\eta_k^{f,i}) & \text{Cov}(\eta_k^{f,i}, \theta_k^{f,i}) \\ \text{Cov}(\theta_k^{f,i}, X_k^{f,i}) & \text{Cov}(\theta_k^{f,i}, \eta_k^{f,i}) & \text{Var}(\theta_k^{f,i}) \end{pmatrix} \begin{pmatrix} H^\top \\ 0_{(n+n_\theta) \times p} \end{pmatrix} + R \end{pmatrix}^{-1} \\ &= \begin{pmatrix} \text{Var}(X_k^{f,i}) H^\top (H \text{Var}(X_k^{f,i}) H^\top + R)^{-1} \\ \text{Cov}(\eta_k^{f,i}, X_k^{f,i}) H^\top (H \text{Var}(X_k^{f,i}) H^\top + R)^{-1} \\ \text{Cov}(\theta_k^{f,i}, X_k^{f,i}) H^\top (H \text{Var}(X_k^{f,i}) H^\top + R)^{-1} \end{pmatrix} := \begin{pmatrix} \widehat{K}_k[X] \\ \widehat{K}_k[\eta] \\ \widehat{K}_k[\theta] \end{pmatrix} \end{aligned} \quad (3.5)$$

$$d_k^i = Y_k + \varepsilon_k^i - \widehat{H} \widehat{X}_k^{f,i} \text{ avec } \varepsilon_k^i \sim \mathcal{N}(0, R) \quad (3.6)$$

$$\widehat{X}_k^{a,i} = \widehat{X}_k^{f,i} + \widehat{K}_k d_k^i \Leftrightarrow \begin{pmatrix} X_k^{a,i} \\ \eta_k^{a,i} \\ \theta_k^{a,i} \end{pmatrix} = \begin{pmatrix} X_k^{f,i} + \widehat{K}_k[X] d_k^i \\ \eta_k^{f,i} + \widehat{K}_k[\eta] d_k^i \\ \theta_k^{f,i} + \widehat{K}_k[\theta] d_k^i \end{pmatrix}. \quad (3.7)$$

Pour que le *state augmentation* soit efficace dans l'estimation de l'erreur de modèle, il faut que le gain de Kalman associé à l'erreur de modèle $\widehat{K}_k[\eta]$ prenne des valeurs non nulles, c'est-à-dire que la matrice de covariance entre les états et les erreurs de modèle prédits $\text{Cov}(\eta_k^{f,i}, X_k^{f,i})$ dans l'équation (3.5) prenne des valeurs non nulles. Ceci est permis par l'utilisation dans l'équation (3.2) du terme aléatoire $\tilde{\eta}_k^i(\theta_{k-1}^{a,i})$ qui dépend des paramètres dynamiques estimés, rendant $\eta_k^{f,i}$ dépendant de $\theta_{k-1}^{a,i}$, comme $X_k^{f,i}$, ce qui renforce ainsi la corrélation entre $\eta_k^{f,i}$ et $X_k^{f,i}$. Le terme aléatoire $\tilde{\eta}_k^i(\theta_{k-1}^{a,i})$ doit être choisi de sorte à ne pas dénaturer la structure spatiale de $\eta_{k-1}^{a,i}$. L'utilisation d'un terme bruité conforme à la physique du système étudié, comme PIME dans le Chapitre 2 pour modéliser l'erreur de

modèle, peut servir pour représenter ce terme aléatoire en tenant compte des paramètres dynamiques. L'efficacité de cette approche est illustrée dans un cas d'étude similaire à celui du Chapitre 2 mais avec un modèle dynamique plus complexe.

3.1 Cas d'étude

On s'intéresse à l'assimilation de données de température d'une barre métallique chauffée en son centre et sujette à une advection périodique au cours du temps. Le vrai état est solution de l'équation d'advection-diffusion inhomogène suivante

$$\frac{\partial X}{\partial t}(x, t) + v(t) \frac{\partial X}{\partial x}(x, t) - \alpha \frac{\partial^2 X}{\partial x^2}(x, t) = r(x, t) \quad (3.8)$$

pour $x \in [0, 1]$ et $t > 0$, avec les conditions initiale et aux bords suivantes

$$X(x, 0) = \sin(\pi x) \quad \text{et} \quad X(0, t) = X(1, t) = 0. \quad (3.9)$$

Le but est de retrouver $X(x, t)$ à partir d'observations partielles et bruitées, sans connaître la source de chaleur extérieure $r(x, t)$ (qui varie aussi en espace contrairement au Chapitre 2), ni le coefficient de diffusion α ni le coefficient d'advection $v(t)$ qui varie en temps.

3.2 Description de l'algorithme utilisé

Comme dans le Chapitre 2, l'espace est discrétisé en n points $\{x_1, \dots, x_n\}$ et on s'intéresse aux instants $t_k = (k-1)dt$ pour $k \in \{1, \dots, K_{final}\}$. L'objectif est d'estimer $X_k = (X(t_k, x_1) \dots X(t_k, x_n))^T$. Pour faire cela, l'EnKF est utilisé pour estimer l'état augmenté

$$\widehat{X}_k = \begin{pmatrix} X_k \\ \eta_k \\ v_k \\ \alpha_k \end{pmatrix} \in \mathbb{R}^{2n+2}$$

où $\eta_k \in \mathbb{R}^n$ représente l'erreur de modèle induite par la méconnaissance de la source de chaleur extérieure $r(x, t)$ et des paramètres dynamiques : $v_k \in \mathbb{R}$ le coefficient d'advection et $\alpha_k \in \mathbb{R}$ le coefficient de diffusion. Le modèle espace-état défini par les équations (3.1a)-(3.1d) est utilisé avec $\theta_k = (v_k \ \alpha_k)^T$,

$C = \begin{pmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\alpha^2 \end{pmatrix}$ et comme modèle dynamique mal spécifié l'équation homogène associée à l'équation (3.8) où $v(t)$ et α ne sont pas connus

$$\frac{\partial X}{\partial t}(x, t) + v(t) \frac{\partial X}{\partial x}(x, t) - \alpha \frac{\partial^2 X}{\partial x^2}(x, t) = 0 \quad (3.10)$$

pour $x \in [0, 1]$ et $t > 0$, avec les conditions initiale et aux bords de l'équation (3.9). En pratique, estimer le terme $M(X_{k-1}; \theta_{k-1})$ dans l'équation (3.1a) revient à calculer la solution à l'instant t_k de l'équation (3.10) discrétisée en temps et en espace en utilisant la méthode de Runge-Kutta d'ordre 4 à pas adaptatif, avec une estimation de X_{k-1} comme condition initiale et une estimation de θ_{k-1} pour les valeurs des paramètres dynamiques. L'EnKF utilisé suit les équations (3.2)-(3.7) où le terme aléatoire $\tilde{\eta}_k^i(\theta_{k-1}^{a,i})$ est déterminé en suivant la même approche que dans la Section 2.4. Ce terme aléatoire correspond alors à une solution bruitée de l'équation stationnaire associée à l'équation (3.8). On résout l'équation stationnaire suivante où le second membre est remplacé par une constante r et où \tilde{v} et $\tilde{\alpha}$ sont deux valeurs inconnues strictement positives des paramètres dynamiques

$$\tilde{v} \frac{\partial X}{\partial x}(x) - \tilde{\alpha} \frac{\partial^2 X}{\partial x^2}(x) = r \quad (3.11)$$

pour $x \in [0, 1]$. Avec les conditions aux bords de l'équation (3.9) on obtient la solution

$$X(x) = \frac{-\frac{r}{\tilde{v}}}{e^{\frac{\tilde{v}x}{\tilde{\alpha}}} - 1} \left(e^{\frac{\tilde{v}x}{\tilde{\alpha}}} - 1 \right) + \frac{rx}{\tilde{v}}. \quad (3.12)$$

r est ensuite remplacé par un bruit blanc $r_k^i \sim \mathcal{N}(0, \sigma_r^2)$ pour $i \in \{1, \dots, N\}$, ce qui permet d'obtenir l'expression du terme aléatoire recherché pour chaque chaque point de la barre x_j avec $j \in \{1, \dots, n\}$

$$\tilde{\eta}_k^i(\tilde{v}, \tilde{\alpha})[j] = \frac{-\frac{r_k^i}{\tilde{v}}}{e^{\frac{\tilde{v}x_j}{\tilde{\alpha}}} - 1} \left(e^{\frac{\tilde{v}x_j}{\tilde{\alpha}}} - 1 \right) + \frac{r_k^i x_j}{\tilde{v}},$$

où \tilde{v} et $\tilde{\alpha}$ sont des valeurs des paramètres dynamiques qui peuvent être approchées par les dernières estimations obtenues par le filtre : $v_{k-1}^{a,i}$ et $\alpha_{k-1}^{a,i}$ pour $i \in \{1, \dots, N\}$. Le terme aléatoire utilisé dans notre algorithme est donc pour $k \in \{1, \dots, K_{final}\}$, $i \in \{1, \dots, N\}$ et $j \in \{1, \dots, n\}$

$$\tilde{\eta}_k^i(\theta_{k-1}^{a,i})[j] = \frac{-\frac{r_k^i}{v_{k-1}^{a,i}}}{e^{\frac{v_{k-1}^{a,i} x_j}{\alpha_{k-1}^{a,i}}} - 1} \left(e^{\frac{v_{k-1}^{a,i} x_j}{\alpha_{k-1}^{a,i}}} - 1 \right) + \frac{r_k^i x_j}{v_{k-1}^{a,i}}. \quad (3.13)$$

L'algorithme utilisé avec PIME est récapitulé dans le Tableau 3.1. La fonction $\max(\tau, \cdot)$, avec $\tau > 0$, est utilisée car les paramètres dynamiques à estimer sont strictement positifs.

TABLE 3.1 – Algorithme avec PIME pour l'estimation en ligne de l'erreur de modèle.

<p><i>Initialisation</i> : pour $i = 1, \dots, N$ généner $v_0^{a,i} \sim \mathcal{U}[v_0^a - \sigma_v, v_0^a + \sigma_v]$ $v_1^{a,i} = v_0^{a,i}$ généner $\alpha_0^{a,i} \sim \mathcal{U}[\alpha_0^a - \sigma_\alpha, \alpha_0^a + \sigma_\alpha]$ $\alpha_1^{a,i} = \alpha_0^{a,i}$ généner $r_1^i \sim \mathcal{N}(0, \sigma_r^2)$ $\eta_1^{a,i} = \tilde{\eta}_1^i(\theta_0^{a,i})$ en suivant l'équation (3.13) $X_1^{a,i} = X_0 + \eta_1^{a,i}$ Pour $k \geq 2$</p> <p><i>Prédiction</i> : pour $i = 1, \dots, N$ $X_k^{f,i} = M(X_{k-1}^{a,i}; \theta_{k-1}^{a,i}) + \eta_{k-1}^{a,i}$ généner $r_k^i \sim \mathcal{N}(0, \sigma_r^2)$ $\eta_k^{f,i} = \eta_{k-1}^{a,i} + \tilde{\eta}_k^i(\theta_{k-1}^{a,i})$ en suivant l'équation (3.13) généner $\tilde{\theta}_k^i \sim \mathcal{N}(0, C)$ $\theta_k^{f,i} = \theta_{k-1}^{a,i} + \tilde{\theta}_k^i$ $\widehat{X}_k^{f,i} = (X_k^{f,i} \quad \eta_k^{f,i} \quad \theta_k^{f,i})^\top$ $\widehat{X}_k^f = \frac{1}{N} \sum_{i=1}^N \widehat{X}_k^{f,i}$ $\widehat{P}_k^f = \frac{1}{N-1} \sum_{i=1}^N (\widehat{X}_k^{f,i} - \widehat{X}_k^f)(\widehat{X}_k^{f,i} - \widehat{X}_k^f)^\top$</p> <p><i>Analyse</i> : pour $i = 1, \dots, N$ $\widehat{H} = (H \quad 0_{p \times (n+2)})$ $\widehat{K}_k = \widehat{P}_k^f \widehat{H}^\top (\widehat{H} \widehat{P}_k^f \widehat{H}^\top + R)^{-1}$ généner $\epsilon_k^i \sim \mathcal{N}(0, R)$ $d_k^i = Y_k + \epsilon_k^i - \widehat{H} \widehat{X}_k^{f,i}$ $\widehat{X}_k^{a,i} = \widehat{X}_k^{f,i} + \widehat{K}_k d_k^i = (X_k^{a,i} \quad \eta_k^{a,i} \quad \theta_k^{a,i})^\top$ $\theta_k^{a,i} = \max(\tau, \theta_k^{a,i})$</p>
--

3.3 Résultats numériques

Notre approche utilisant PIME pour l'estimation en ligne de l'erreur de modèle est comparée, toujours dans le cadre du *state augmentation*, à l'utilisation d'un bruit blanc Gaussien avec une matrice de covariance bien calibrée comme QSS dans le Chapitre 2. L'algorithme utilisant ce bruit blanc Gaussien est détaillé dans le Tableau 3.2 avec la matrice de covariance associée pour $1 \leq j, j' \leq n$

$$Q[j, j'] = \sigma^2 e^{-\lambda |x_j - x_{j'}|},$$

où les paramètres σ et λ sont optimisés à la main. Cet algorithme est basé sur le même modèle espace-état que notre algorithme en Section 3.2 avec cependant $\eta_k = \eta_{k-1} + \tilde{\eta}_k$ où le bruit blanc Gaussien utilisé est $\tilde{\eta}_k \sim \mathcal{N}(0, Q)$.

TABLE 3.2 – Algorithme avec bruit blanc Gaussien pour l'estimation en ligne de l'erreur de modèle.

<p><i>Initialisation</i> : pour $i = 1, \dots, N$ généner $v_1^{a,i} \sim \mathcal{U}[v_0^a - \sigma_v, v_0^a + \sigma_v]$ généner $\alpha_1^{a,i} \sim \mathcal{U}[\alpha_0^a - \sigma_\alpha, \alpha_0^a + \sigma_\alpha]$ généner $\eta_1^{a,i} \sim \mathcal{N}(0, Q)$ $X_1^{a,i} = X_0 + \eta_1^{a,i}$ Pour $k \geq 2$ <i>Prédiction</i> : pour $i = 1, \dots, N$ $X_k^{f,i} = M(X_{k-1}^{a,i}, \theta_{k-1}^{a,i}) + \eta_{k-1}^{a,i}$ généner $\tilde{\eta}_k^i \sim \mathcal{N}(0, Q)$ $\eta_k^{f,i} = \eta_{k-1}^{a,i} + \tilde{\eta}_k^i$ généner $\tilde{\theta}_k^i \sim \mathcal{N}(0, C)$ $\theta_k^{f,i} = \theta_{k-1}^{a,i} + \tilde{\theta}_k^i$ $\widehat{X}_k^{f,i} = (X_k^{f,i} \quad \eta_k^{f,i} \quad \theta_k^{f,i})^\top$ $\widehat{X}_k^f = \frac{1}{N} \sum_{i=1}^N \widehat{X}_k^{f,i}$ $\widehat{P}_k^f = \frac{1}{N-1} \sum_{i=1}^N (\widehat{X}_k^{f,i} - \widehat{X}_k^f)(\widehat{X}_k^{f,i} - \widehat{X}_k^f)^\top$ <i>Analyse</i> : pour $i = 1, \dots, N$ $\widehat{H} = (H \quad 0_{p \times (n+2)})$ $\widehat{K}_k = \widehat{P}_k^f \widehat{H}^\top (\widehat{H} \widehat{P}_k^f \widehat{H}^\top + R)^{-1}$ généner $\epsilon_k^i \sim \mathcal{N}(0, R)$ $d_k^i = Y_k + \epsilon_k^i - \widehat{H} \widehat{X}_k^{f,i}$ $\widehat{X}_k^{a,i} = \widehat{X}_k^{f,i} + \widehat{K}_k d_k^i = (X_k^{a,i} \quad \eta_k^{a,i} \quad \theta_k^{a,i})^\top$ $\theta_k^{a,i} = \max(\tau, \theta_k^{a,i})$</p>
--

Les deux algorithmes décrits dans les Tableaux 3.1 et 3.2 sont comparés sur leur capacité à retrouver le vrai état en estimant à la fois l'erreur de modèle et les paramètres dynamiques. Le vrai état X_k pour $k \in \{1, \dots, K_{final}\}$ est obtenu avec le vrai modèle dynamique décrit par l'équation (3.8) où les coefficients de diffusion et d'advection à estimer sont respectivement $\alpha = 0.05$ et $v(t) = (\sin(\pi t))^2 + 0.1$. Aussi la source de chaleur extérieure inconnue est $r(x, t) = \sin(\pi x) \sin(t)$ de sorte à ce que la barre métallique soit alternativement chauffée et refroidie en son centre au cours du temps. Un point de la barre sur deux est observé avec en plus une erreur d'observation représentée par un bruit blanc Gaussien avec matrice de covariance R connue. Les paramètres communs aux deux algorithmes sont présentés dans le Tableau 3.3. Comme dans le Chapitre 2, les valeurs des paramètres $\sigma_r = 0.009$ pour notre méthode ainsi que $\sigma = 0.007$ et $\lambda = 0.01$ pour l'autre méthode sont obtenues en minimisant la RMSE globale définie dans l'équation (2.14). Il en est de même pour obtenir les valeurs des écarts-types σ_v et σ_α dans le Tableau 3.3.

Paramètre	Valeur
K_{final}	100
n	100
p	50
dt	0.1
N	30
R	$0.01I_p$
σ_v	0.1
σ_α	10^{-3}
v_0^a	0.1
α_0^a	0.06

TABLE 3.3 – Valeurs des paramètres communs aux deux algorithmes.

L'évolution temporelle de la température de la barre obtenue par les deux algorithmes est comparée à celle du vrai état X_k et des observations Y_k dans la Figure 3.1. Pour chaque algorithme, la moyenne empirique des analyses $X_k^a = \frac{1}{N} \sum_{i=1}^N X_k^{a,i}$ est utilisée à chaque itération k pour obtenir l'image associée. Les images obtenues sont globalement proches de celle du vrai état, de plus l'algorithme avec PIME semble identifier les pics de température (en rouge) aux instants $t = 1, 2, 8$ contrairement à l'algorithme avec bruit Gaussien. La RMSE globale est de 0.028 pour l'algorithme avec PIME et de 0.038 pour l'algorithme avec bruit Gaussien.

L'évolution temporelle de l'erreur de modèle η_k estimée par les deux algorithmes est présentée dans la Figure 3.2. L'image de l'estimation de l'erreur de modèle obtenue par l'algorithme avec PIME semble plus cohérente avec l'image du vrai état montrée dans la Figure 3.1 : on retrouve l'advection, la diffusion et la source de chaleur extérieure. Ceci est permis par l'utilisation dans notre algorithme du terme aléatoire $\tilde{\eta}_k^i(\theta_{k-1}^{a,i})$ qui tient compte des paramètres dynamiques, rendant l'état et l'erreur de modèle plus corrélés. L'estimation de l'erreur de modèle obtenue par l'algorithme avec bruit Gaussien semble ne pas tenir compte de l'advection ni de la diffusion, mais parvient aussi à identifier l'évolution temporelle de la source de chaleur extérieure, ce qui montre l'efficacité de l'estimation en ligne de l'erreur de modèle par *state augmentation* lorsque l'erreur de modèle est structurée temporellement.

L'estimation des paramètres dynamiques v_k et α_k par les deux algorithmes est ensuite montrée sur la Figure 3.3. Aussi, pour chaque algorithme, la RMSE globale associée au paramètre d'advection et celle associée au paramètre de diffusion sont respectivement calculées suivant

$$\frac{1}{K_{final}} \sum_{k=1}^{K_{final}} \sqrt{\frac{1}{N} \sum_{i=1}^N (v_k^{a,i} - v_k)^2}$$

$$\text{et } \frac{1}{K_{final}} \sum_{k=1}^{K_{final}} \sqrt{\frac{1}{N} \sum_{i=1}^N (\alpha_k^{a,i} - \alpha_k)^2}.$$

La probabilité de couverture de chaque algorithme est aussi calculée pour chaque paramètre. Elle représente la probabilité que la vraie valeur du paramètre dynamique se trouve dans l'intervalle de confiance 95% obtenu par l'algorithme à chaque instant. Sa valeur optimale est 0.95. La RMSE globale donne des informations sur la moyenne des membres générés par l'algorithme et la probabilité de couverture sur la dispersion de ces membres. Les résultats obtenus pour chaque paramètre sont présentés dans les Tableaux 3.4 et 3.5. Ces derniers montrent, comme la Figure 3.3, que les algorithmes parviennent à retrouver les paramètres dynamiques, avec une meilleure estimation pour l'algorithme avec PIME.

	RMSE globale	Probabilité de couverture
Algorithme avec PIME	0.250	0.72
Algorithme avec bruit Gaussien	0.284	0.71

TABLE 3.4 – RMSE globale et probabilité de couverture de chaque algorithme pour le paramètre d'advection $v(t)$.

	RMSE globale	Probabilité de couverture
Algorithme avec PIME	0.005	0.84
Algorithme avec bruit Gaussien	0.010	0.31

TABLE 3.5 – RMSE globale et probabilité de couverture de chaque algorithme pour le paramètre de diffusion α .

On s'intéresse désormais au milieu de la barre, l'évolution temporelle de la température en ce point $X_k[n/2]$ estimée par les deux algorithmes est représentée sur la Figure 3.4. Aussi, comme précédemment, la RMSE globale et la probabilité de couverture associées au milieu de la barre sont données pour chaque algorithme dans le Tableau 3.6. Ce dernier, comme la Figure 3.4, montre que l'algorithme avec PIME est plus performant.

	RMSE globale	Probabilité de couverture
Algorithme avec PIME	0.030	0.68
Algorithme avec bruit Gaussien	0.043	0.48

TABLE 3.6 – RMSE globale et probabilité de couverture de chaque algorithme pour le milieu de la barre.

En se focalisant maintenant à $t = 8$, on compare la température estimée en chaque point de la barre par les deux algorithmes dans la Figure 3.5. L'algorithme avec PIME semble mieux retrouver le vrai état X_{80} , en décrivant une

évolution spatiale de la température plus lisse et plus conforme à celle du vrai état que l'algorithme avec bruit Gaussien. Cette différence s'explique par l'utilisation du terme aléatoire $\tilde{\eta}_{80}^i(\theta_{79}^{a,i})$, dans l'algorithme avec PIME, construit à partir de la solution stationnaire de l'équation (3.12) dont la courbe est représentée sur la Figure 3.6 avec les vraies valeurs des paramètres dynamiques à $t = 8$. Cette courbe a une allure proche de celle du vrai état X_{80} dans la Figure 3.5. Le terme aléatoire $\tilde{\eta}_{80}^i(\theta_{79}^{a,i})$ permet alors une estimation de l'erreur de modèle plus conforme à la physique du système étudié, qui mène à une meilleure estimation de X_{80} par l'algorithme avec PIME.

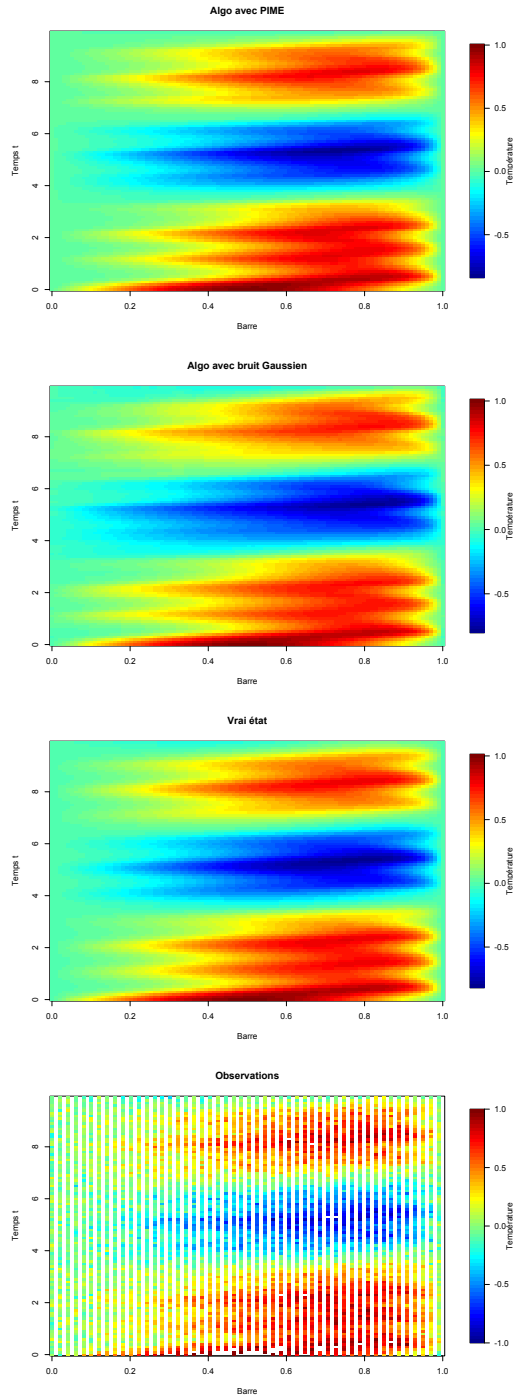


FIGURE 3.1 – Images de l'évolution du système obtenue par chaque algorithme jusqu'à $t = 10$ (i.e. $k = 100$).

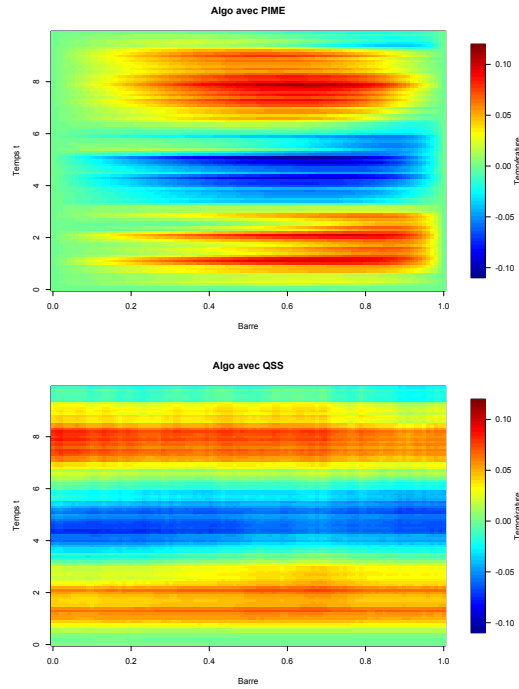


FIGURE 3.2 – Images de l'évolution de l'erreur de modèle estimée par chaque algorithme. Pour chaque algorithme, la moyenne empirique des analyses obtenues pour l'erreur de modèle $\eta_k^a = \frac{1}{N} \sum_{i=1}^N \eta_k^{a,i}$ est utilisée à chaque itération k .

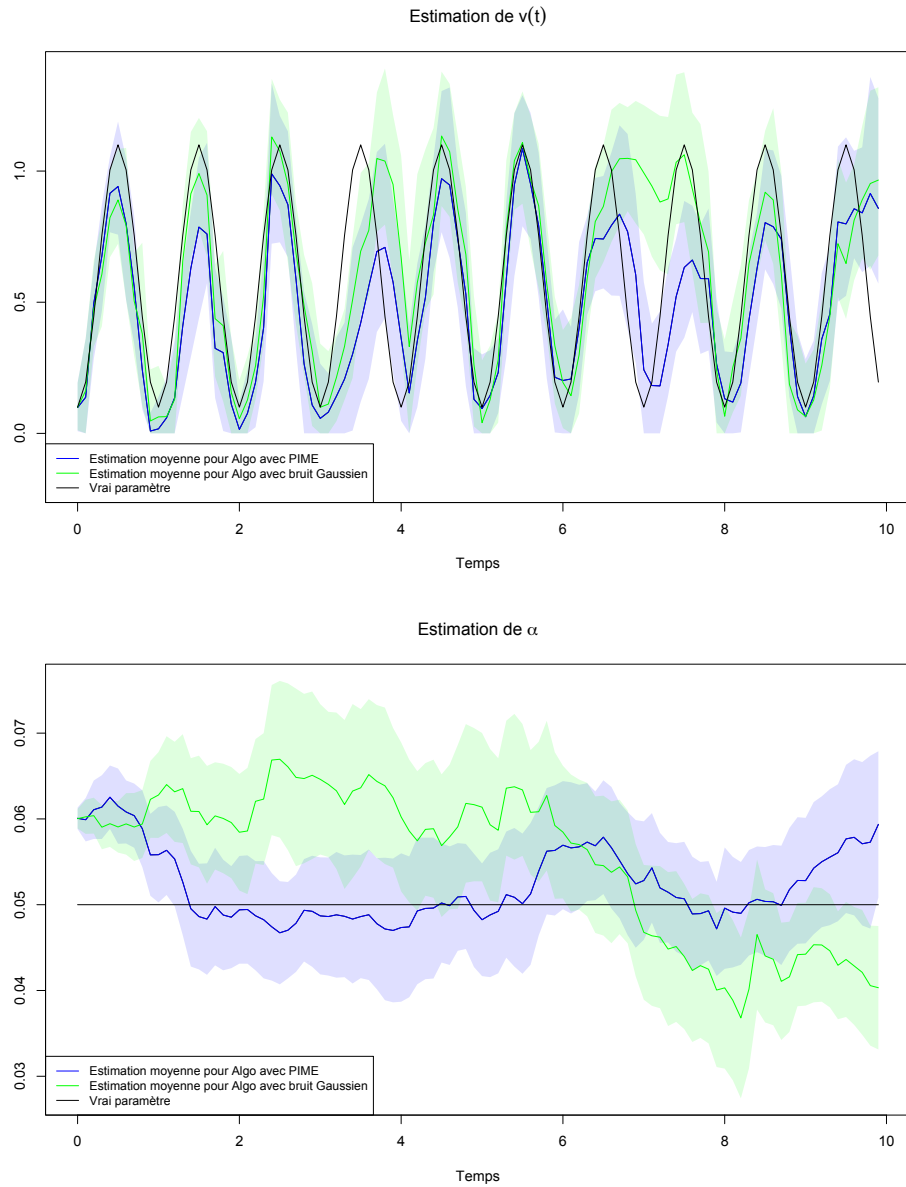


FIGURE 3.3 – Estimation de $v(t)$ et α par les deux algorithmes. Pour chaque paramètre dynamique, l'intervalle de confiance 95% centré en la moyenne est représenté pour chaque algorithme.

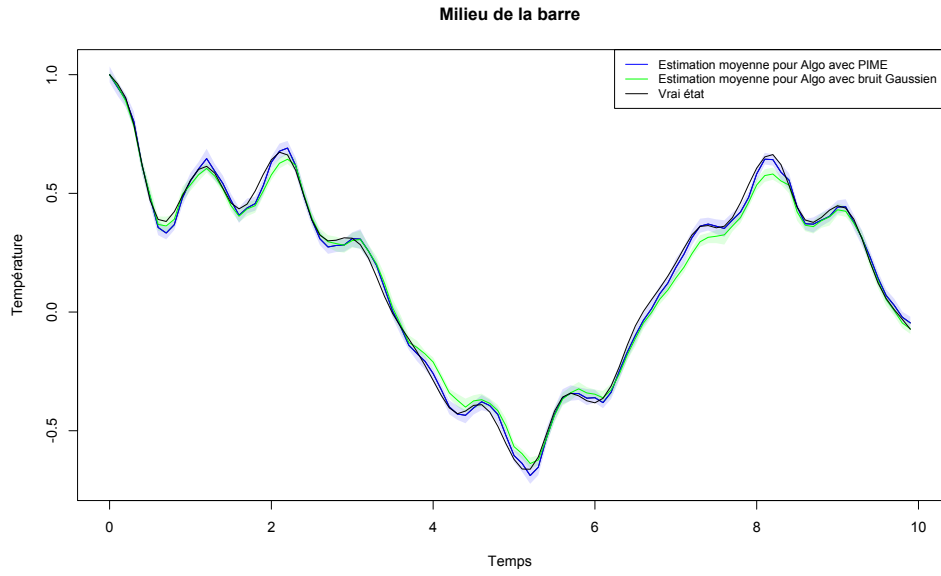


FIGURE 3.4 – Estimation de $X_k[n/2]$ par les deux algorithmes avec leur intervalle de confiance 95% centré en la moyenne.

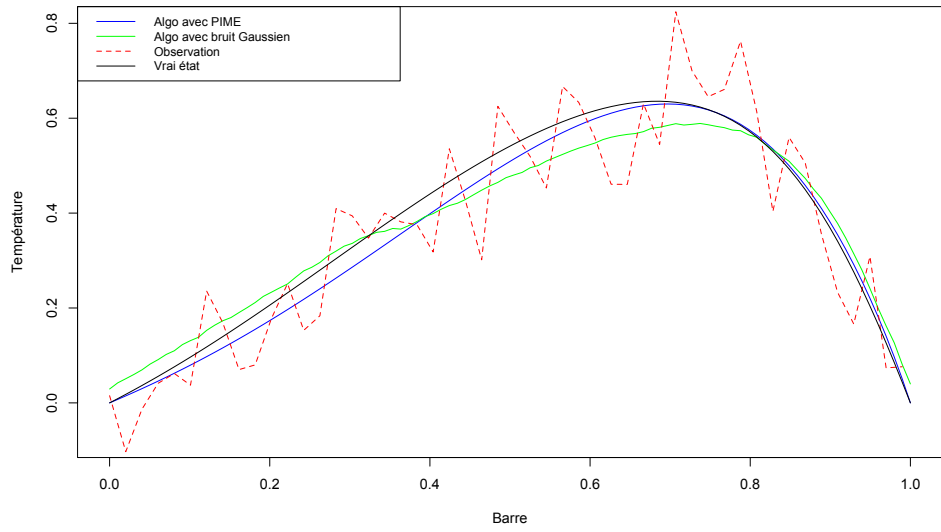


FIGURE 3.5 – Estimation de X_{80} par les deux algorithmes en utilisant leur moyenne empirique des analyses X_{80}^a .

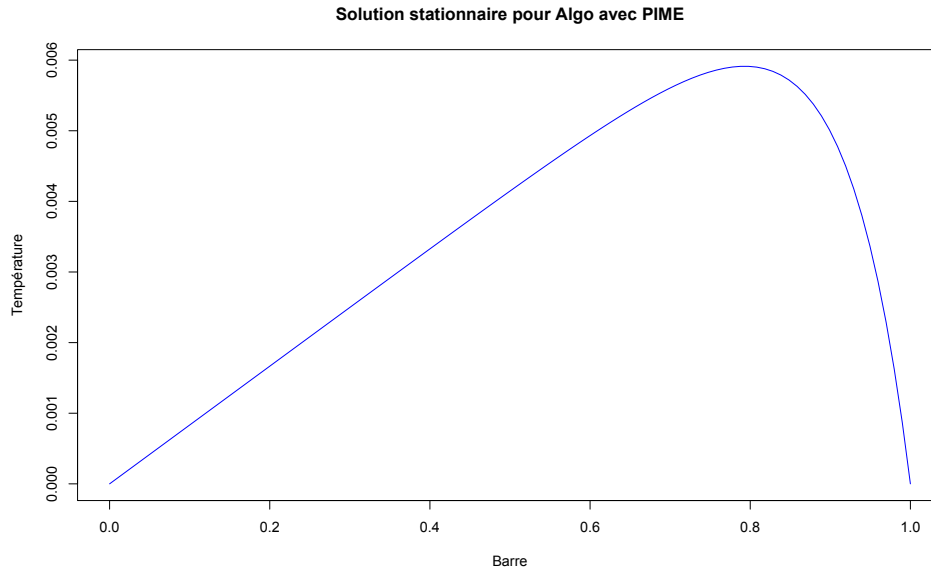


FIGURE 3.6 – Solution stationnaire décrite par l'équation (3.12) avec $\tilde{v} = v(8)$, $\tilde{\alpha} = \alpha$ et $r = 0.005$.

3.4 Conclusion

Une nouvelle approche a été détaillée dans ce chapitre pour quantifier l'incertitude liée au modèle dynamique. Cette méthode consiste à estimer conjointement l'état, l'erreur de modèle et des paramètres dynamiques en pratiquant la *state augmentation* avec l'EnKF. Cependant le terme aléatoire dans la marche aléatoire, utilisée pour propager l'erreur de modèle à l'instant suivant, restait à déterminer. L'idée alors développée pour représenter ce terme aléatoire a été d'utiliser la méthode de l'erreur de modèle inspirée par la physique (PIME) décrite dans le Chapitre 2, de sorte à guider l'apprentissage de l'erreur de modèle avec l'EnKF en tenant compte des paramètres dynamiques. Cette méthode a ensuite été comparée à l'utilisation d'un bruit blanc Gaussien avec matrice de covariance paramétrée. Le modèle-jouet utilisé pour l'expérience correspond à une équation inhomogène d'advection-diffusion dont le second membre et les paramètres d'advection et de diffusion sont inconnus. Ces paramètres dynamiques ont, comme l'erreur de modèle, été rajoutés à l'état augmenté pour être estimés. Les résultats numériques ont montré que la *state augmentation* permet une estimation en ligne pertinente de l'erreur de modèle, surtout pour l'algorithme avec PIME qui permet de tenir compte des paramètres dynamiques et d'avoir une estimation conforme à la physique du système étudié. Les résultats numériques ont aussi montré que l'algorithme avec PIME retrouve mieux le vrai état et les paramètres dynamiques que l'algorithme utilisant un bruit blanc Gaussien.

Cette nouvelle approche se révèle être pertinente pour corriger un modèle dynamique mal spécifié avec des paramètres dynamiques inconnus.

Contributions :

- Nouvelle approche pour estimer en ligne l'erreur de modèle,
- Cette méthode permet de réduire significativement l'erreur de modèle notamment lorsque des paramètres dynamiques sont inconnus.

Avant-propos du Chapitre 4

La méthode détaillée dans le Chapitre 3 nécessite d'optimiser des paramètres stochastiques à la main. Ces paramètres ne peuvent être correctement estimés en ligne par *state augmentation* avec l'EnKF comme illustré par [15] car ils ne sont pas corrélés à l'état. Une autre possibilité est d'utiliser le filtre particulaire (PF) pour estimer des paramètres stochastiques. Ainsi par *state augmentation*, l'EnKF peut servir à estimer l'état tandis que le PF estime les paramètres stochastiques. Cette nouvelle combinaison de l'EnKF avec le PF est présentée ci-après pour l'estimation en ligne de paramètres stochastiques du schéma d'assimilation, dans le but de quantifier les incertitudes liées au modèle dynamique et aux observations.

Chapitre 4

State and Stochastic Parameters Estimation with Combined Ensemble Kalman and Particle Filters

Jules Guillot, Pierre Ailliot, Emmanuel Frénod, Juan Ruiz and Pierre Tandeo

Abstract

In data assimilation, quantifying the uncertainties related to the dynamical model and the observations is of key importance to retrieve the latent true state. These uncertainties are usually represented as Gaussian white noises with covariance matrices depending on parameters to estimate. The new approach described in this paper allows to retrieve the true state using state augmentation with the unknown time-dependent stochastic parameters. Both state and parameters are estimated online by combining an ensemble Kalman filter with a particle filter. The ensemble Kalman filter is usually used for high-dimensional geophysical states whereas the particle filter is suitable for nonlinear dynamical models in low-dimensional systems. The strengths of these two methods are gathered in the proposed algorithm, where the ensemble Kalman filter generates the members that estimate the true state whereas the particle filter generates the particles that estimate the set of unknown stochastic parameters. The generic method is first detailed to estimate the model or observation error. Then it is developed to simultaneously estimate the parameters of the covariance inflation and localization methods that allow to overcome the uncertainties in the data assimilation system. Experiments with the Lorenz-96 model are led to compare the new algorithm with state-of-the-art methods. The results show the ability of the new method to retrieve the true state and to estimate the time-dependent stochastic parameters. The algorithm can be easily built from an existing en-

semble Kalman filter with low additional cost and without further running the dynamical model.

4.1 Introduction

Data assimilation consists in combining a dynamical model with observation data to retrieve the latent true state of a system. The dynamical model can be based on physical assumptions, this is for the model-driven approach (see [11]), or fully based on the observations, this is for the data-driven approach (see [34]). In the model-driven approach, the dynamical model is generally misspecified because of unknown physical phenomena as in [26] or unknown parameters for instance, generating an error called the model error. Also, the observations are often noisy because of imperfect sensors, generating an error with respect to the true state that is called the measurement error. Moreover, part of the discrepancies between the dynamical model and the observations, as those arising from the different resolved spatio-temporal scales, are referred to as representation errors (see [31]). The observation error is composed of the measurement and representation errors. Quantifying the model and observation errors for data assimilation remains a difficult task as explained in [46]. These two uncertainties are usually represented as Gaussian white noises whose the covariance matrices \mathbf{Q} (for the model error) and \mathbf{R} (for the observation error) depend on stochastic parameters to estimate. This estimation problem has been addressed by many authors and the existing methods can be classified in different categories detailed hereinafter.

The grid based search method used in [48] is probably the most obvious way to estimate each stochastic parameter : a fixed list of possible values for the parameter is used to find the optimal value that usually minimizes the error with respect to the true state. The main drawback of this method is its high computational cost since the filter has to be run many times.

Another approach, based on the ensemble Kalman filter (EnKF) developed by [17], consists in using an ensemble of EnKFs, each of them running with different parameters values. This has been done by [43] to estimate the parameters of the covariance matrix of a Gaussian white noise which is directly added inside the dynamical model to treat the model error, this is a stochastic parameterization strategy as detailed in [38] and [8]. These nested EnKFs allow to estimate different types of parameters but it is costly.

In a different way, the expectation-maximization (EM) methods based on the maximum likelihood have been developed for offline [47] and online [13] estimation. The offline version can estimate \mathbf{Q} and \mathbf{R} but with an high computational cost (because of the use of the ensemble Kalman smoother, see [18]) whereas the online version can jointly estimate \mathbf{Q} and \mathbf{R} with a reasonable computational cost.

The state augmentation, widely used in data assimilation, consists in augmenting the state with the unknown parameters and assuming that these latter follow a simple dynamical model (e.g., a random walk). This method is easy to

implement and suitable for the online estimation of dynamical parameters as in [3]. However it is not appropriated for the estimation of stochastic parameters as shown in [15] when using the EnKF, because stochastic parameters are not correlated with the true state. In order to circumvent this issue and instead of using the EnKF for both the state and stochastic parameters, it has been proposed to use the EnKF for the state and the particle filter (PF), see [49], for the stochastic parameters. The EnKF is suitable for a high-dimensional state and the PF for a low number of stochastic parameters because of the curse of dimensionality discussed in [44]. Different combinations of the EnKF with the PF have been studied as in [22] to estimate the parameters of \mathbf{R} or in [45] to estimate the parameters of \mathbf{Q} or \mathbf{R} . These methods use the PF to generate the particles that estimate the set of stochastic parameters, combined with the EnKF to generate each member related to a particle for the state estimation, with a reasonable computational cost.

Jointly estimating \mathbf{Q} and \mathbf{R} remains difficult as shown by [46] because both have an impact on the ensemble spread. That is why instead of directly estimating \mathbf{Q} and \mathbf{R} , stress can be laid on the covariance matrix of the forecasted members. The covariance inflation of [6] consists in multiplying this covariance matrix by an inflation parameter to reproduce the effect of the model or observation error on it. This strategy also allows to reduce the impact of the sampling error, which is due to the use of a too small ensemble size for computational reasons when running the dynamical model, even if the well-defined \mathbf{Q} and \mathbf{R} matrices are used in the data assimilation system. The inflation parameter is usually estimated online using the innovation statistics of [16]. Another combination of the PF with the EnKF has also been elaborated by [2] to estimate this parameter where it is supposed to be static. The covariance inflation is generally combined with the covariance localization of [29] to further mitigate the impact of the sampling error and avoid the filter degeneracy. The covariance localization allows to eliminate the long-range spurious correlations in the covariance matrix of the forecasted members which is rank-deficient. For that, a mask depending on a localization parameter is applied to this covariance matrix. Usually a grid search is used to estimate the localization parameter but it is expensive.

For all the works cited above, there are no numerical results when the parameters to estimate of \mathbf{Q} and \mathbf{R} are time-dependent, that is to say when we focus on \mathbf{Q}_t and \mathbf{R}_t . In this paper, for the state and stochastic parameters estimation, a new combination of the PF with the EnKF that boosts the synergy between them, called the PF-EnKF algorithm, is detailed for estimating \mathbf{Q}_t , \mathbf{R}_t or both time-dependent inflation and localization parameters. At each time step, a set of particles is first generated to represent the distribution of the stochastic parameters, then one step of the EnKF is run for each particle, finally the generated members are combined using the usual weighting/resampling steps of the PF, as for the particles. The main contribution of this algorithm is its ability to estimate time-dependent stochastic parameters of the data assimilation system, being easy to implement with a reasonable computational cost compared to existing methods. Moreover, its genericity allows to elaborate an adaptive approach to jointly perform inflation and localization.

The rest of the paper is organized as follows. The new method is developed in Section 4.2 for different case studies. Numerical results with the Lorenz-96 model are then discussed in Section 4.3. Finally, concluding remarks and perspectives are given in Section 4.4.

4.2 The new PF-EnKF method

The work is based on the following state-space model for $t \in \{1, \dots, T\}$:

$$\begin{cases} \mathbf{x}_t = M(\mathbf{x}_{t-1}) + \boldsymbol{\eta}_t & \text{with } \boldsymbol{\eta}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t), & (4.1a) \\ \mathbf{y}_t = \mathbf{H}\mathbf{x}_t + \boldsymbol{\varepsilon}_t & \text{with } \boldsymbol{\varepsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t). & (4.1b) \end{cases}$$

\mathbf{x}_t is the latent true state in \mathbb{R}^n and \mathbf{y}_t is the related observation in \mathbb{R}^p . $M(\cdot)$ is the dynamical model and \mathbf{H} is the linear observation operator. $\boldsymbol{\eta}_t$ is the model error and $\boldsymbol{\varepsilon}_t$ is the observation error. In this paper $\mathbf{Q}_t = \mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t})$ and $\mathbf{R}_t = \mathbf{R}(\boldsymbol{\theta}_{\mathbf{R},t})$ with the sets of stochastic parameters $\boldsymbol{\theta}_{\mathbf{Q},t}$ and $\boldsymbol{\theta}_{\mathbf{R},t}$.

4.2.1 PF-EnKF for estimating the model or observation error

In this section, we first assume that the set of stochastic parameters $\boldsymbol{\theta}_{\mathbf{R},t}$ related to the observation error is known and we discuss the online estimation of $\boldsymbol{\theta}_{\mathbf{Q},t}$ related to the model error. The estimation of $\boldsymbol{\theta}_{\mathbf{R},t}$ when $\boldsymbol{\theta}_{\mathbf{Q},t}$ is known is similar and detailed in Appendix 4.5.1.

$\boldsymbol{\theta}_{\mathbf{Q},t}$ is supposed to be a Markov process as it is usually done when using state-augmentation. In terms of conditional distributions, the state-space model can be written as

$$\begin{aligned} p(\boldsymbol{\theta}_{\mathbf{Q},t} | \boldsymbol{\theta}_{\mathbf{Q},1:t-1}, \mathbf{x}_{1:t-1}) &\sim p(\boldsymbol{\theta}_{\mathbf{Q},t} | \boldsymbol{\theta}_{\mathbf{Q},t-1}), \\ p(\mathbf{x}_t | \mathbf{x}_{1:t-1}, \boldsymbol{\theta}_{\mathbf{Q},1:t}) &\sim \phi(\mathbf{x}_t; M(\mathbf{x}_{t-1}), \mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t})), \\ p(\mathbf{y}_t | \mathbf{x}_{1:t}, \boldsymbol{\theta}_{\mathbf{Q},1:t}) &\sim \phi(\mathbf{y}_t; \mathbf{H}\mathbf{x}_t, \mathbf{R}_t), \end{aligned}$$

with ϕ the probability density function of the Gaussian distribution and $p(\boldsymbol{\theta}_{\mathbf{Q},t} | \boldsymbol{\theta}_{\mathbf{Q},t-1})$ a transition kernel which describes the evolution of the set of unknown stochastic parameters (e.g., a random walk).

The goal is to estimate \mathbf{x}_t and $\boldsymbol{\theta}_{\mathbf{Q},t}$ at each time t knowing the observations $\mathbf{y}_{1:t}$. This relies on

$$p(\mathbf{x}_t, \boldsymbol{\theta}_{\mathbf{Q},t} | \mathbf{y}_{1:t}) = p(\mathbf{x}_t | \boldsymbol{\theta}_{\mathbf{Q},t}, \mathbf{y}_{1:t}) p(\boldsymbol{\theta}_{\mathbf{Q},t} | \mathbf{y}_{1:t}), \quad (4.2)$$

where the conditional distribution $p(\mathbf{x}_t | \boldsymbol{\theta}_{\mathbf{Q},t}, \mathbf{y}_{1:t})$ is estimated using the EnKF and $p(\boldsymbol{\theta}_{\mathbf{Q},t} | \mathbf{y}_{1:t})$ using the PF. The equations related to each filter are detailed below.

EnKF for $p(\mathbf{x}_t|\boldsymbol{\theta}_{Q,t}, \mathbf{y}_{1:t})$:

For the forecast step at time t , N_{memb} state estimates, called the forecasted members, are generated by applying the dynamical model to the analyzed members at time $t-1$ denoted $\mathbf{x}_{t-1}^{a,i}$ and adding a random perturbation depending on $\boldsymbol{\theta}_{Q,t}$ to obtain for $i \in \{1, \dots, N_{memb}\}$

$$\mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}) = M(\mathbf{x}_{t-1}^{a,i}) + \boldsymbol{\eta}_t^i \quad \text{with } \boldsymbol{\eta}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}_{Q,t})), \quad (4.3)$$

then the empirical mean and covariance matrix of the forecasted members are computed

$$\mathbf{x}_t^f(\boldsymbol{\theta}_{Q,t}) = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}), \quad (4.4)$$

$$\mathbf{P}_t^f(\boldsymbol{\theta}_{Q,t}) = \frac{1}{N_{memb} - 1} \sum_{i=1}^{N_{memb}} \left(\mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}) - \mathbf{x}_t^f(\boldsymbol{\theta}_{Q,t}) \right) \left(\mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}) - \mathbf{x}_t^f(\boldsymbol{\theta}_{Q,t}) \right)^\top. \quad (4.5)$$

For the analysis step, each forecasted member is corrected using the available observation \mathbf{y}_t as in [10] to generate the analyzed members for $i \in \{1, \dots, N_{memb}\}$

$$\mathbf{x}_t^{a,i}(\boldsymbol{\theta}_{Q,t}) = \mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}) + \mathbf{K}_t(\boldsymbol{\theta}_{Q,t}) \left(\mathbf{y}_t + \boldsymbol{\varepsilon}_t^i - \mathbf{H} \mathbf{x}_t^{f,i}(\boldsymbol{\theta}_{Q,t}) \right) \quad (4.6)$$

with

$$\mathbf{K}_t(\boldsymbol{\theta}_{Q,t}) = \mathbf{P}_t^f(\boldsymbol{\theta}_{Q,t}) \mathbf{H}^\top \left(\mathbf{H} \mathbf{P}_t^f(\boldsymbol{\theta}_{Q,t}) \mathbf{H}^\top + \mathbf{R}_t \right)^{-1} \quad (4.7)$$

and

$$\boldsymbol{\varepsilon}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t). \quad (4.8)$$

PF for $p(\boldsymbol{\theta}_{Q,t}|\mathbf{y}_{1:t})$:

The theorem of Bayes shows that

$$p(\boldsymbol{\theta}_{Q,t}|\mathbf{y}_{1:t}) \propto p(\boldsymbol{\theta}_{Q,t}|\mathbf{y}_{1:t-1}) p(\mathbf{y}_t|\mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{Q,t}). \quad (4.9)$$

Using a sequential importance resampling (SIR) scheme, the conditional distribution $p(\boldsymbol{\theta}_{Q,t}|\mathbf{y}_{1:t-1})$ is approximated by the discrete distribution

$$p(\boldsymbol{\theta}_{Q,t}|\mathbf{y}_{1:t-1}) \approx \frac{1}{N_\theta} \sum_{j=1}^{N_\theta} \delta_{\boldsymbol{\theta}_{Q,t}^{f,j}}(\boldsymbol{\theta}_{Q,t})$$

where $\delta(\cdot)$ denotes the Dirac distribution. The forecasted particles $\boldsymbol{\theta}_{Q,t}^{f,j}$, for $j \in \{1, \dots, N_\theta\}$, are generated from the conditional distribution $p(\boldsymbol{\theta}_{Q,t}|\boldsymbol{\theta}_{Q,t-1}^{a,j})$ where $\boldsymbol{\theta}_{Q,t-1}^{a,j}$ denotes the j -th analyzed particle at $t-1$.

The previous EnKF scheme allows to approximate the likelihood in Eq. (4.9) using

$$p(\mathbf{y}_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{\mathbf{Q},t}) = \int p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_{1:t-1}, \boldsymbol{\theta}_{\mathbf{Q},t}) d\mathbf{x}_t.$$

Moreover, based on [40], Eq. (4.9) can be approximated by the discrete distribution

$$p(\boldsymbol{\theta}_{\mathbf{Q},t} | \mathbf{y}_{1:t}) \approx \sum_{j=1}^{N_\theta} \gamma_t^j \delta_{\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}}(\boldsymbol{\theta}_{\mathbf{Q},t}) \quad (4.10)$$

where $\gamma_t^j = \frac{\phi(\mathbf{y}_t; \mathbf{H}\mathbf{x}_t^f(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}), \mathbf{H}\mathbf{P}_t^f(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j})\mathbf{H}^\top + \mathbf{R}_t)}{\sum_{k=1}^{N_\theta} \phi(\mathbf{y}_t; \mathbf{H}\mathbf{x}_t^f(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,k}), \mathbf{H}\mathbf{P}_t^f(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,k})\mathbf{H}^\top + \mathbf{R}_t)}$ for $j \in \{1, \dots, N_\theta\}$.

$$(4.11)$$

Thus following Eq. (4.2), the PF and EnKF have to be combined.

PF-EnKF algorithm :

For the forecast step at time t , the previous members $\mathbf{x}_{t-1}^{a,i}$ are propagated by the dynamical model to obtain the propagated members

$$\mathbf{x}_t^{p,i} = M(\mathbf{x}_{t-1}^{a,i}) \quad \text{for } i \in \{1, \dots, N_{memb}\},$$

with their empirical mean and covariance matrix

$$\mathbf{x}_t^p = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{p,i}$$

$$\mathbf{P}_t^p = \frac{1}{N_{memb} - 1} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)(\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)^\top.$$

It is not required to run the dynamical model more times than with the usual EnKF, this does not increase the computational cost compared to the usual EnKF, which can be useful for the numerical weather prediction community when the models are expensive.

Then Eqs. (4.3-4.8) of the EnKF are used to estimate the true state by replacing the unknown set of stochastic parameters $\boldsymbol{\theta}_{\mathbf{Q},t}$ by a forecasted particle $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$. The EnKF is run for each particle $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$. More precisely, for $i \in \{1, \dots, N_{memb}\}$ and $j \in \{1, \dots, N_\theta\}$, the forecasted members are given by

$$\mathbf{x}_t^{f,i,j} = \mathbf{x}_t^{p,i} + \mathbf{C}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}) \boldsymbol{\xi}_t^i \quad (4.12)$$

with the random perturbation $\mathbf{C}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}) \boldsymbol{\xi}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}))$, where $\boldsymbol{\xi}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with the identity matrix \mathbf{I} and $\mathbf{C}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j})$ comes from the Cholesky decomposition of $\mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j})$ so that $\mathbf{C}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j})\mathbf{C}^\top(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}) = \mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j})$.

Then instead of computing the empirical mean and covariance matrix of the forecasted members, we use the empirical mean and covariance matrix of the propagated members with the theoretical mean and covariance matrix of the random perturbations to obtain

$$\mathbf{x}_t^f = \mathbf{x}_t^p, \quad (4.13)$$

$$\mathbf{P}_t^{f,j} = \mathbf{P}_t^p + \mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}) \quad \text{for } j \in \{1, \dots, N_\theta\}. \quad (4.14)$$

The numerical experiments that we performed show that the formulation of the random perturbation in Eq. (4.12) combined with the estimates of the mean and covariance matrix of the forecasted members in Eqs. (4.13)-(4.14), improve the performance of the algorithm by reducing the impact of the sampling error in the likelihood used to weight each particle, which allows to better identify the relevant particles.

The analyzed members are then computed for $i \in \{1, \dots, N_{memb}\}$ and $j \in \{1, \dots, N_\theta\}$

$$\mathbf{x}_t^{a,i,j} = \mathbf{x}_t^{f,i,j} + \mathbf{K}_t^j \mathbf{d}_t^{i,j},$$

using the Kalman gain $\mathbf{K}_t^j = \mathbf{P}_t^{f,j} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \mathbf{R}_t)^{-1}$ and the innovation $\mathbf{d}_t^{i,j} = \mathbf{y}_t + \boldsymbol{\varepsilon}_t^i - \mathbf{H} \mathbf{x}_t^{f,i,j}$ with $\boldsymbol{\varepsilon}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_t)$.

The weight of each particle $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$ is obtained for $j \in \{1, \dots, N_\theta\}$ using Eq. (4.11)

$$\gamma_t^j = \frac{\phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^f, \mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \mathbf{R}_t)}{\sum_{k=1}^{N_\theta} \phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^f, \mathbf{H} \mathbf{P}_t^{f,k} \mathbf{H}^\top + \mathbf{R}_t)}. \quad (4.15)$$

In the analysis step, the i -th member has N_θ versions $\mathbf{x}_t^{a,i,j}$ related to each particle $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$. To have N_{memb} members instead of $N_{memb} \times N_\theta$ to avoid the running of the dynamical model too many times at $t+1$, the weighted average of the versions of the i -th member is computed using the weights of the particles $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$ to obtain the averaged members

$$\mathbf{x}_t^{a,i} = \sum_{j=1}^{N_\theta} \gamma_t^j \mathbf{x}_t^{a,i,j} \quad \text{for } i \in \{1, \dots, N_{memb}\}.$$

The forecasted particles $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j}$ are resampled using the discrete distribution in Eq. (4.10) with the weights from Eq. (4.15) to give the analyzed particles $\boldsymbol{\theta}_{\mathbf{Q},t}^{a,j}$ for $j \in \{1, \dots, N_\theta\}$. These latter will be used at $t+1$ to generate the forecasted particles $\boldsymbol{\theta}_{\mathbf{Q},t+1}^{f,j}$ from the conditional distribution $p(\boldsymbol{\theta}_{\mathbf{Q},t+1}^{f,j} | \boldsymbol{\theta}_{\mathbf{Q},t}^{a,j})$.

The PF-EnKF algorithm for estimating the model error is detailed in Algorithm 4 and a cycle at time t of it is schematized in Figure 4.1. This algorithm can be easily implemented from an existing EnKF.

When both $\theta_{Q,t}$ and $\theta_{R,t}$ are unknown, their simultaneous estimation with our method is complicated because the likelihood used depends on both Q_t and R_t , making it difficult to identify relevant particles that estimate both $\theta_{Q,t}$ and $\theta_{R,t}$. Instead of estimating these latter, the algorithm detailed in Section 4.2.2 overcomes the lack of knowledge on both the model and observation errors by practising covariance inflation and localization, jointly estimating the related parameters.

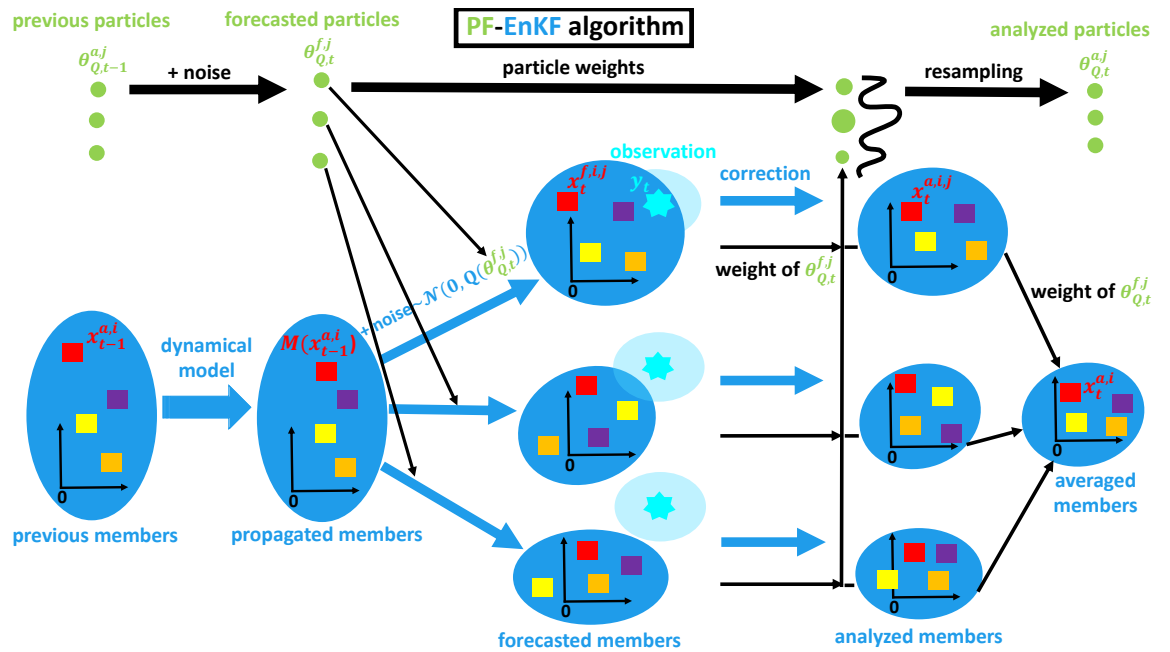


FIGURE 4.1 – PF-EnKF cycle to estimate Q_t .

Algorithm 4 PF-EnKF for estimating Q_t

Require: $Q(\cdot)$, R_t and $\theta_{Q,0}$

- 1: *Initialization :*
- 2: **for** $i = 1, \dots, N_{memb}$ **do**
- 3: $\eta_1^i \sim \mathcal{N}(\mathbf{0}, Q(\theta_{Q,0}))$
- 4: $\mathbf{x}_1^{a,i} = \mathbf{x}_0 + \eta_1^i$
- 5: **end for**
- 6: **for** $j = 1, \dots, N_\theta$ **do**
- 7: $\theta_{Q,1}^{a,j} \sim p(\theta_{Q,1} | \theta_{Q,0})$
- 8: **end for**
- 9: **for** $t = 2, \dots, T$ **do**
- 10: *Forecast :*
- 11: **for** $i = 1, \dots, N_{memb}$ **do**
- 12: $\mathbf{x}_t^{p,i} = M(\mathbf{x}_{t-1}^{a,i})$
- 13: $\xi_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 14: $\varepsilon_t^i \sim \mathcal{N}(\mathbf{0}, R_t)$
- 15: **end for**
- 16: $\mathbf{x}_t^p = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{p,i}$
- 17: $\mathbf{P}_t^p = \frac{1}{N_{memb}-1} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)(\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)^\top$
- 18: **for** $j = 1, \dots, N_\theta$ **do**
- 19: $\theta_{Q,t}^{f,j} \sim p(\theta_{Q,t} | \theta_{Q,t-1}^{a,j})$
- 20: Cholesky decomposition of $Q(\theta_{Q,t}^{f,j})$ to obtain $C(\theta_{Q,t}^{f,j})$
- 21: $\mathbf{P}_t^{f,j} = \mathbf{P}_t^p + Q(\theta_{Q,t}^{f,j})$
- 22: $\mathbf{K}_t^j = \mathbf{P}_t^{f,j} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + R_t)^{-1}$
- 23: **for** $i = 1, \dots, N_{memb}$ **do**
- 24: $\mathbf{x}_t^{f,i,j} = \mathbf{x}_t^{p,i} + C(\theta_{Q,t}^{f,j}) \xi_t^i$
- 25: *Analysis :*
- 26: $\mathbf{d}_t^{i,j} = \mathbf{y}_t + \varepsilon_t^i - \mathbf{H} \mathbf{x}_t^{f,i,j}$
- 27: $\mathbf{x}_t^{a,i,j} = \mathbf{x}_t^{f,i,j} + \mathbf{K}_t^j \mathbf{d}_t^{i,j}$
- 28: **end for**
- 29: **end for**
- 30: **for** $j = 1, \dots, N_\theta$ **do**
- 31: $\gamma_t^j = \frac{\phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^p, \mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + R_t)}{\sum_{k=1}^{N_\theta} \phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^p, \mathbf{H} \mathbf{P}_t^{f,k} \mathbf{H}^\top + R_t)}$
- 32: **end for**
- 33: **for** $i = 1, \dots, N_{memb}$ **do**
- 34: $\mathbf{x}_t^{a,i} = \sum_{j=1}^{N_\theta} \gamma_t^j \mathbf{x}_t^{a,i,j}$
- 35: **end for**
- 36: resample the forecasted particles $\theta_{Q,t}^{f,j}$ knowing the weights γ_t^j to obtain the analyzed particles $\theta_{Q,t}^{a,j}$
- 37: **end for**

4.2.2 PF-EnKF with inflation and localization

Here both the model and observation errors are unknown but the following approach can also be applied when they are known. To compensate the lack of knowledge on the model and observation errors to retrieve \mathbf{x}_t , we focus on the covariance matrix of the forecasted members \mathbf{P}_t^f , because both \mathbf{Q}_t and \mathbf{R}_{t-1} have an impact on \mathbf{P}_t^f . The inflation of this latter is led to quantify the model and observation uncertainties and to avoid the filter degeneracy. Moreover, the localization of \mathbf{P}_t^f is used to reduce the impact of the sampling error due to the use of a small ensemble size N_{memb} , as explained by [5]. The resulting approach is detailed in Section 4.2.2 and an illustrative example is given in Section 4.2.2.

Algorithm for the joint estimation of the inflation and localization parameters

The inflation and localization of \mathbf{P}_t^f are led jointly by the Schur product $\mathbf{L}(\boldsymbol{\theta}_{\mathbf{L},t}) \circ \mathbf{P}_t^f$ which is the element-wise multiplication between $\mathbf{L}(\boldsymbol{\theta}_{\mathbf{L},t})$ and \mathbf{P}_t^f . With the set of unknown stochastic parameters $\boldsymbol{\theta}_{\mathbf{L},t} = (\lambda_{\mathbf{L},t}, l_{\mathbf{L},t})$ and $\mathbf{L}(\cdot)$ which is a n by n positive semi-definite matrix that represents the decay of correlations in the physical space. Each element of $\mathbf{L}(\boldsymbol{\theta}_{\mathbf{L},t})$ is computed using the Gaspari-Cohn (GC) correlation function in [23] that depends on the localization parameter $l_{\mathbf{L},t}$ and inflated by $\lambda_{\mathbf{L},t}$, so that $\mathbf{L}(\boldsymbol{\theta}_{\mathbf{L},t})[k, k'] = \lambda_{\mathbf{L},t} \text{GC}(l_{\mathbf{L},t})[k, k']$ for $(k, k') \in \{1, \dots, n\}^2$. This correlation function is a polynomial approximation of a Gaussian density but with compact support that depends on $l_{\mathbf{L},t}$, allowing to remove the long-range spurious correlations in \mathbf{P}_t^f to reduce the sampling error impact. The goal is to estimate $\boldsymbol{\theta}_{\mathbf{L},t}$ using the same methodology as in Section 4.2.1. The resulting PF-EnKF algorithm jointly estimates time-dependent inflation and localization parameters. It is detailed in Algorithm 5 and a cycle at time t of it is schematized in Figure 4.2. This algorithm can use a priori estimates of \mathbf{Q}_t and \mathbf{R}_t , respectively denoted as $\widehat{\mathbf{Q}}_t$ and $\widehat{\mathbf{R}}_t$, to generate the random perturbations used to compute the members. Also, because covariance inflation and localization are used here, there is no need to use the theoretical mean and covariance matrix of the forecasted members in Eqs. (16-17) for reducing the sampling error as in Section 4.2.1.

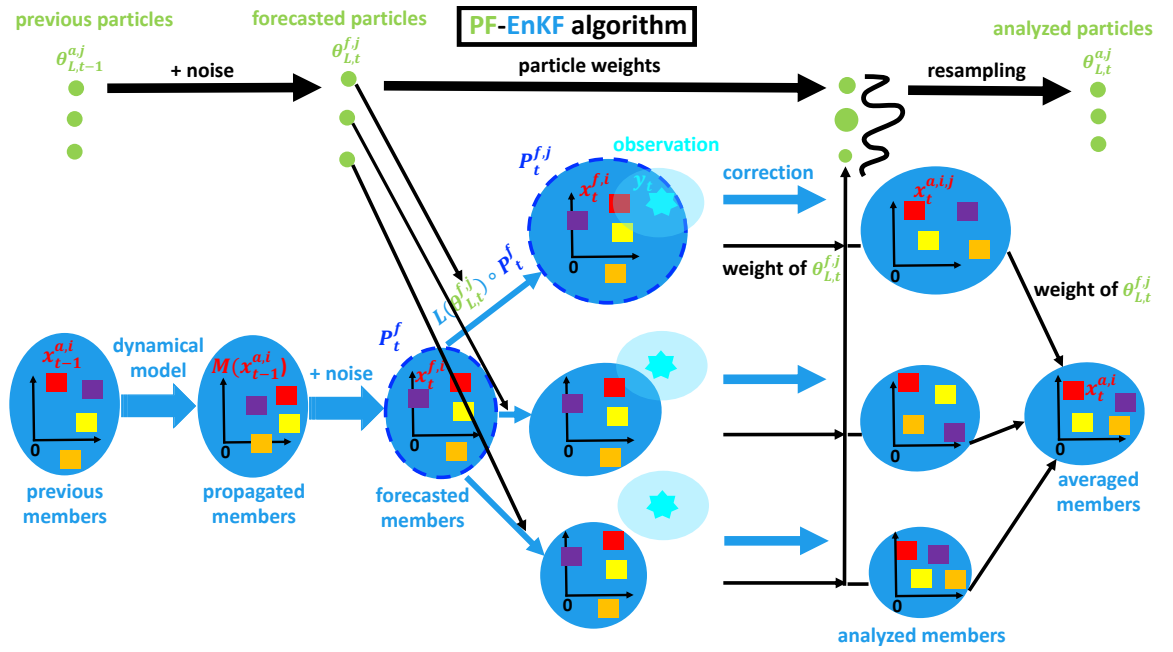


FIGURE 4.2 – PF-EnKF cycle with inflation and localization.

Algorithm 5 PF-EnKF with inflation and localization, when \mathbf{Q}_t and \mathbf{R}_t are unknown

Require: $\widehat{\mathbf{Q}}_t$, $\widehat{\mathbf{R}}_t$, $L(\cdot)$ and $\boldsymbol{\theta}_{L,0}$

- 1: *Initialization :*
- 2: **for** $i = 1, \dots, N_{memb}$ **do**
- 3: $\boldsymbol{\eta}_1^i \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{Q}}_1)$
- 4: $\mathbf{x}_1^{a,i} = \mathbf{x}_0 + \boldsymbol{\eta}_1^i$
- 5: **end for**
- 6: **for** $j = 1, \dots, N_\theta$ **do**
- 7: $\boldsymbol{\theta}_{L,1}^{a,j} \sim p(\boldsymbol{\theta}_{L,1} | \boldsymbol{\theta}_{L,0})$
- 8: **end for**
- 9: **for** $t = 2, \dots, T$ **do**
- 10: *Forecast :*
- 11: **for** $i = 1, \dots, N_{memb}$ **do**
- 12: $\boldsymbol{\eta}_t^i \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{Q}}_t)$
- 13: $\boldsymbol{\varepsilon}_t^i \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{R}}_t)$
- 14: $\mathbf{x}_t^{f,i} = M(\mathbf{x}_{t-1}^{a,i}) + \boldsymbol{\eta}_t^i$
- 15: **end for**
- 16: $\mathbf{x}_t^f = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{f,i}$
- 17: $\mathbf{P}_t^f = \frac{1}{N_{memb}-1} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)(\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)^\top$
- 18: **for** $j = 1, \dots, N_\theta$ **do**
- 19: $\boldsymbol{\theta}_{L,t}^{f,j} \sim p(\boldsymbol{\theta}_{L,t} | \boldsymbol{\theta}_{L,t-1}^{a,j})$
- 20: $\mathbf{P}_t^{f,j} = L(\boldsymbol{\theta}_{L,t}^{f,j}) \circ \mathbf{P}_t^f$
- 21: $\mathbf{K}_t^j = \mathbf{P}_t^{f,j} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \widehat{\mathbf{R}}_t)^{-1}$
- 22: *Analysis :*
- 23: **for** $i = 1, \dots, N_{memb}$ **do**
- 24: $\mathbf{d}_t^i = \mathbf{y}_t + \boldsymbol{\varepsilon}_t^i - \mathbf{H} \mathbf{x}_t^{f,i}$
- 25: $\mathbf{x}_t^{a,i,j} = \mathbf{x}_t^{f,i} + \mathbf{K}_t^j \mathbf{d}_t^i$
- 26: **end for**
- 27: **end for**
- 28: **for** $j = 1, \dots, N_\theta$ **do**
- 29: $\gamma_t^j = \frac{\phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^f, \mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \widehat{\mathbf{R}}_t)}{\sum_{k=1}^{N_\theta} \phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^f, \mathbf{H} \mathbf{P}_t^{f,k} \mathbf{H}^\top + \widehat{\mathbf{R}}_t)}$
- 30: **end for**
- 31: **for** $i = 1, \dots, N_{memb}$ **do**
- 32: $\mathbf{x}_t^{a,i} = \sum_{j=1}^{N_\theta} \gamma_t^j \mathbf{x}_t^{a,i,j}$
- 33: **end for**
- 34: resample the forecasted particles $\boldsymbol{\theta}_{L,t}^{f,j}$ knowing the weights γ_t^j to obtain the analyzed particles $\boldsymbol{\theta}_{L,t}^{a,j}$
- 35: **end for**

Illustrative example

The stochastic parameters are estimated using the likelihood in Eq. 29. To stress the importance of using the likelihood as a metric, a simple study is led by comparing the use of the likelihood to the use of the root mean square error (RMSE) for estimating both the inflation and localization parameters.

To do so, we focus on the following state-space model for $t \in \{1, \dots, T = 10^4\}$:

$$\begin{cases} \mathbf{x}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{P}), \\ \mathbf{y}_t \sim \mathcal{N}(\mathbf{H}\mathbf{x}_t, \mathbf{R}). \end{cases}$$

The true state \mathbf{x}_t is in \mathbb{R}^n and its $n = 20$ components correspond to equally spaced grid points on a circle. It is a Gaussian white noise with covariance matrix

$$\mathbf{P}[k, k'] = \alpha e^{-\frac{d(k, k')}{\beta}} \text{ for } (k, k') \in \{1, \dots, n\}^2$$

with $d(k, k')$ the distance between two grid points on the circle, $\alpha = 2$ and $\beta = 5$. One component of \mathbf{x}_t on two is observed such as \mathbf{y}_t is in \mathbb{R}^{10} and made noisy by an observation error that depends on $\mathbf{R} = \mathbf{I}$.

In our experiment summarized in Algorithm 6, we consider that α is unknown and $\widehat{\mathbf{P}}$ is used instead of \mathbf{P} to generate the members that estimate \mathbf{x}_t , with

$$\widehat{\mathbf{P}}[k, k'] = e^{-\frac{d(k, k')}{\beta}} \text{ for } (k, k') \in \{1, \dots, n\}^2.$$

Covariance inflation is led to compensate the misspecification of $\widehat{\mathbf{P}}$. Moreover, the ensemble size $N_{memb} = 20$ is small, so covariance localization is used to reduce the sampling error impact. The covariance inflation and localization are practiced jointly following the strategy shown in Section 4.2.2 through $\mathbf{L}(\boldsymbol{\theta}_{L,t}) \circ \mathbf{P}_t^f$. At each time t the same grids of possible values for the unknown stochastic parameters are used: $\{1, 1.25, \dots, 3\}$ for the inflation parameter $\lambda_{L,t}$ and $\{2, 3, \dots, 10\}$ for the localization parameter $l_{L,t}$. $\boldsymbol{\theta}_{L,t}$ is thus estimated by $N_{\boldsymbol{\theta}} = 81$ couples $\boldsymbol{\theta}_{L,t}^j$ that correspond to the different combinations of the possible values of the stochastic parameters. Then the log-likelihood and RMSE related to each couple $\boldsymbol{\theta}_{L,t}^j$ are computed in Eqs. (11-12) at every time t . The log-likelihood is used here instead of the likelihood to better differentiate the couples. Finally, for each $\boldsymbol{\theta}_{L,t}^j$, the temporal means of the log-likelihood and RMSE are computed, allowing to obtain Figure 4.3. The best couples of parameters values that maximize the likelihood and minimize the RMSE are close to each other. The likelihood is able to identify the same interesting couples as the ones obtained with the RMSE while it is based on partial and noisy observations, contrary to the RMSE that requires the true state. This shows that the likelihood is a relevant metric for estimating both the inflation and localization parameters.

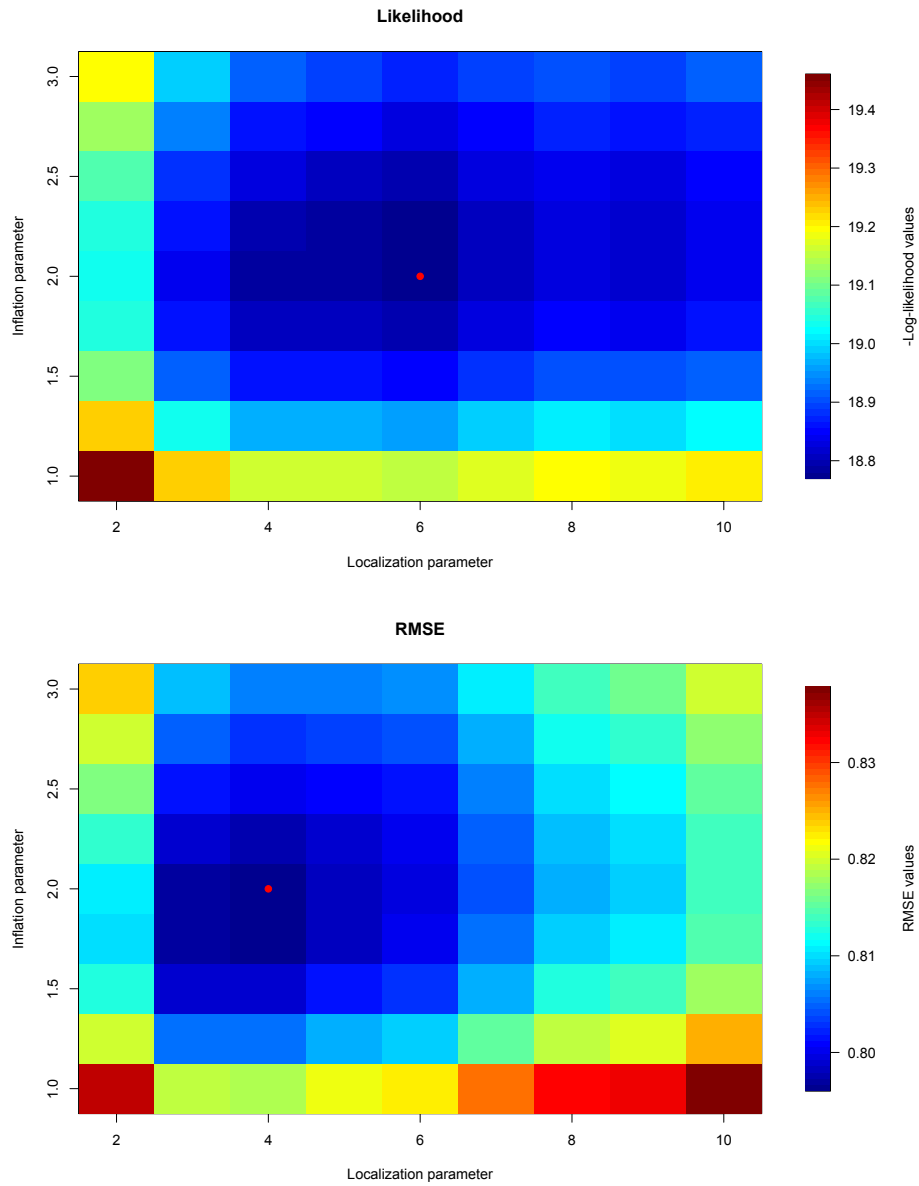


FIGURE 4.3 – Comparison of the couples of parameters values (each represented by a pixel) using the likelihood (on the top) and the RMSE (on the bottom). The best couple is identified by a red point for each method.

Algorithm 6 Estimation of the inflation and localization parameters using the likelihood and RMSE

Require: $\widehat{\mathbf{P}}$ and \mathbf{R}

```

1: for  $t = 1, \dots, T$  do
2:   for  $i = 1, \dots, N_{memb}$  do
3:      $\mathbf{x}_t^{f,i} \sim \mathcal{N}(\mathbf{0}, \widehat{\mathbf{P}})$ 
4:   end for
5:    $\mathbf{x}_t^f = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{f,i}$ 
6:    $\mathbf{P}_t^f = \frac{1}{N_{memb}-1} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)(\mathbf{x}_t^{f,i} - \mathbf{x}_t^f)^\top$ 
7:   for  $j = 1, \dots, N_\theta$  do
8:      $\mathbf{P}_t^{f,j} = \mathbf{L}(\boldsymbol{\theta}_t^j) \circ \mathbf{P}_t^f$ 
9:      $\mathbf{K}_t^j = \mathbf{P}_t^{f,j} \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \mathbf{R})^{-1}$ 
10:     $\mathbf{x}_t^{a,j} = \mathbf{x}_t^f + \mathbf{K}_t^j (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t^f)$ 
11:    Likelihood $_t^j = \log(\phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^f, \mathbf{H} \mathbf{P}_t^{f,j} \mathbf{H}^\top + \mathbf{R}))$ 
12:    RMSE $_t^j = \sqrt{\frac{1}{n} (\mathbf{x}_t^{a,j} - \mathbf{x}_t)^\top (\mathbf{x}_t^{a,j} - \mathbf{x}_t)}$ 
13:   end for
14: end for
15: for  $j = 1, \dots, N_\theta$  do
16:   Likelihood $^j = \frac{1}{T} \sum_{t=1}^T \text{Likelihood}_t^j$ 
17:   RMSE $^j = \frac{1}{T} \sum_{t=1}^T \text{RMSE}_t^j$ 
18: end for

```

4.3 Numerical results

The PF-EnKF algorithm is tested with the Lorenz-96 dynamical model for estimating \mathbf{Q}_t (or \mathbf{R}_t in Appendix 4.5.1) and then both the inflation and localization parameters when \mathbf{Q}_t and \mathbf{R}_t are unknown and a small ensemble size N_{memb} is used. It is compared to the EnKF that uses the true \mathbf{Q}_t in the first case study and with the combination of traditional inflation and localization methods in the second one.

The Lorenz-96 model (see 36) is a one-dimensional model, representing the evolution of a meteorological quantity in n sectors of a latitude circle. It is defined for $k \in \{1, \dots, n = 40\}$ by the ordinary differential equation

$$\frac{\partial \mathbf{x}_{t,k}}{\partial t} = (\mathbf{x}_{t,k+1} - \mathbf{x}_{t,k-2}) \mathbf{x}_{t,k-1} - \mathbf{x}_{t,k} + F$$

with periodic boundary conditions. The time step is 0.05 and $F = 8$ to have a chaotic behavior.

The true state \mathbf{x}_t is generated following Eq. (4.1a) for $t \in \{1, \dots, T = 500\}$ with

$$\mathbf{Q}_t[k, k'] = \lambda_{\mathbf{Q},t}^2 e^{-\frac{d(k,k')^2}{l_{\mathbf{Q},t}^2}} \quad \text{for } (k, k') \in \{1, \dots, n\}^2.$$

$d(k, k')$ is the distance between two grid points on the circle, $\lambda_{\mathbf{Q},t}$ and $l_{\mathbf{Q},t}$ are respectively the variance and spatial scale parameters of \mathbf{Q}_t . The values on the diagonal of \mathbf{Q}_t are equal to $\lambda_{\mathbf{Q},t}^2$ and the more the elements of the matrix are away from the diagonal, the more the value of these elements is close to 0, with a speed that depends on $l_{\mathbf{Q},t}$.

The initial true state is $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. One grid point on two is observed (so $p = 20$) and the observation \mathbf{y}_t is generated following Eq. (4.1b) with

$$\mathbf{R}_t[k, k'] = \lambda_{\mathbf{R},t}^2 e^{-\frac{d(k,k')^2}{l_{\mathbf{R},t}^2}} \quad \text{for } (k, k') \in \{1, \dots, p\}^2. \quad (4.17)$$

The sets of stochastic parameters of \mathbf{Q}_t and \mathbf{R}_t are respectively $\boldsymbol{\theta}_{\mathbf{Q},t} = (\lambda_{\mathbf{Q},t}, l_{\mathbf{Q},t})$ and $\boldsymbol{\theta}_{\mathbf{R},t} = (\lambda_{\mathbf{R},t}, l_{\mathbf{R},t})$.

4.3.1 Results for the estimation of the model error

The PF-EnKF algorithm detailed in Section 4.2.1 is used with $N_{memb} = 100$, $N_\theta = 100$ and $\mathbf{R}_t = 0.1\mathbf{I}$ (which corresponds to a simple form of Eq. 4.17 with $\lambda_{\mathbf{R},t} = \frac{\sqrt{10}}{10}$ and $l_{\mathbf{R},t}$ tends to 0, for each time t).

At $t = 1$, the members are generated following $\mathbf{x}_1^{a,i} = \mathbf{x}_0 + \boldsymbol{\eta}_1^i$ for $i \in \{1, \dots, N_{memb}\}$, where $\boldsymbol{\eta}_1^i$ is a Gaussian white noise with covariance matrix $\mathbf{Q}(\boldsymbol{\theta}_{\mathbf{Q},0})$ where $\boldsymbol{\theta}_{\mathbf{Q},0} = (\lambda_{\mathbf{Q},0}, l_{\mathbf{Q},0}) = (1, 1)$ and for $j \in \{1, \dots, N_\theta\}$ we have $\boldsymbol{\theta}_{\mathbf{Q},1}^{a,j} = (\lambda_{\mathbf{Q},1}^{a,j}, l_{\mathbf{Q},1}^{a,j})$ with $\lambda_{\mathbf{Q},1}^{a,j}$ and $l_{\mathbf{Q},1}^{a,j}$ respectively following the uniform distributions with strictly positive support $\mathcal{U}[0, 2\lambda_{\mathbf{Q},0}]$ and $\mathcal{U}[0, 2l_{\mathbf{Q},0}]$.

For $t \geq 2$, the forecasted particles $\boldsymbol{\theta}_{\mathbf{Q},t}^{f,j} = (\lambda_{\mathbf{Q},t}^{f,j}, l_{\mathbf{Q},t}^{f,j})$ are computed using the random walks $\lambda_{\mathbf{Q},t}^{f,j} = \max(\lambda_{\mathbf{Q},t-1}^{a,j} + \tilde{\lambda}_{\mathbf{Q},t}^j, v_{min})$ and $l_{\mathbf{Q},t}^{f,j} = \max(l_{\mathbf{Q},t-1}^{a,j} + \tilde{l}_{\mathbf{Q},t}^j, v_{min})$ for $j \in \{1, \dots, N_\theta\}$, with the additive noises $\tilde{\lambda}_{\mathbf{Q},t}^j \sim \mathcal{N}(0, \sigma_{\lambda, \mathbf{Q}}^2)$ and $\tilde{l}_{\mathbf{Q},t}^j \sim \mathcal{N}(0, \sigma_{l, \mathbf{Q}}^2)$ to avoid the degeneracy problem of the PF when the particles concentrate on a single particle over time. The hyperparameters are set to $\sigma_{\lambda, \mathbf{Q}} = 0.1$ and $\sigma_{l, \mathbf{Q}} = 0.1$, their tuning has a low impact on the RMSE as shown in Appendix 4.5.2. Because the variance and spatial scale parameters of \mathbf{Q}_t are strictly positive, we constraint the values of the forecasted particles to be larger than $v_{min} = 10^{-4}$.

The goal is to retrieve \mathbf{x}_t and the parameters of \mathbf{Q}_t for $t \in \{1, \dots, T\}$

$$\lambda_{\mathbf{Q},t} = 1 + 0.5 \sin\left(\frac{t}{10}\right),$$

$$l_{\mathbf{Q},t} = \sqrt{3 + 2 \cos\left(\frac{t}{20}\right)}.$$

The temporal evolution of \mathbf{x}_t and \mathbf{y}_t is shown in Figure 4.4.

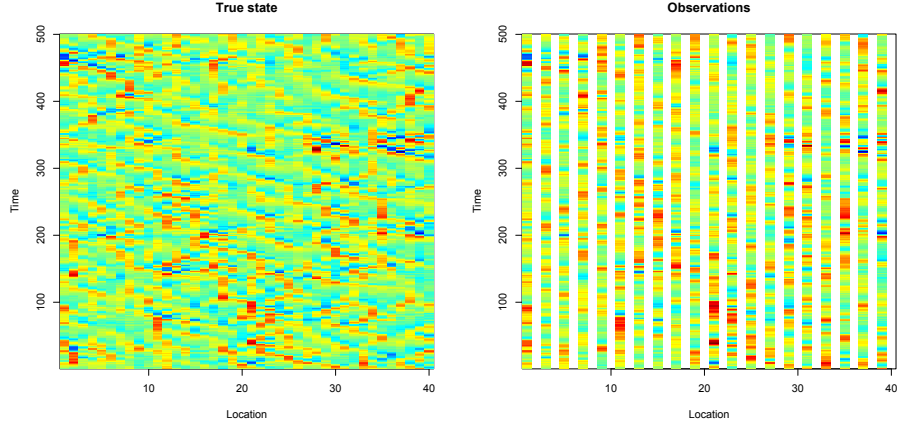


FIGURE 4.4 – True spatio-temporal evolution of the system (on the left) with the partial and noisy observation data (on the right).

Using the sample mean of the particle distribution, $\lambda_{\mathbf{Q},t}$ and $l_{\mathbf{Q},t}$ are respectively estimated by $\frac{1}{N_{\theta}} \sum_{j=1}^{N_{\theta}} \lambda_{\mathbf{Q},t}^{a,j}$ and $\frac{1}{N_{\theta}} \sum_{j=1}^{N_{\theta}} l_{\mathbf{Q},t}^{a,j}$. The temporal evolution of the true parameters is compared with the one of the estimates obtained by the PF-EnKF algorithm in Figure 4.5. This latter shows that our method is able to retrieve the true stochastic parameters, even if $\theta_{\mathbf{Q},0}$ is ill-chosen, with a better accuracy for the variance parameter $\lambda_{\mathbf{Q},t}$. Note that the delay for the estimation of the spatial scale parameter $l_{\mathbf{Q},t}$ is due to the low number of observed grid points $p = 20$. This delay is reduced and the fitting with $l_{\mathbf{Q},t}$ is improved when p is increased as shown in Figure 4.6.

The PF-EnKF can also retrieve the parameters of \mathbf{R}_t (see Appendix 4.5.1).

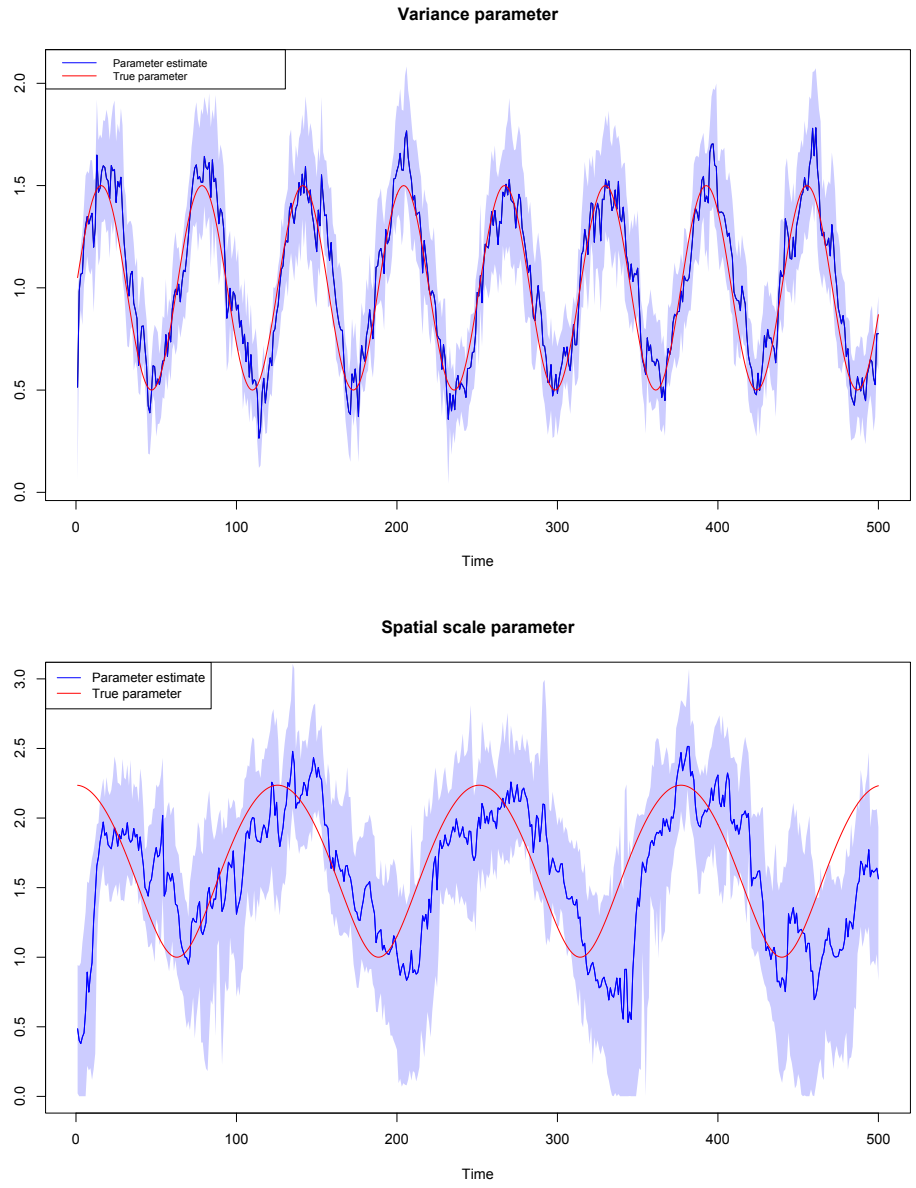


FIGURE 4.5 – Estimation of the variance parameter $\lambda_{Q,t}$ and spatial scale parameter $l_{Q,t}$ with the PF-EnKF algorithm. For each panel, the 95% confidence interval is plotted using the quantiles of the particle distribution.

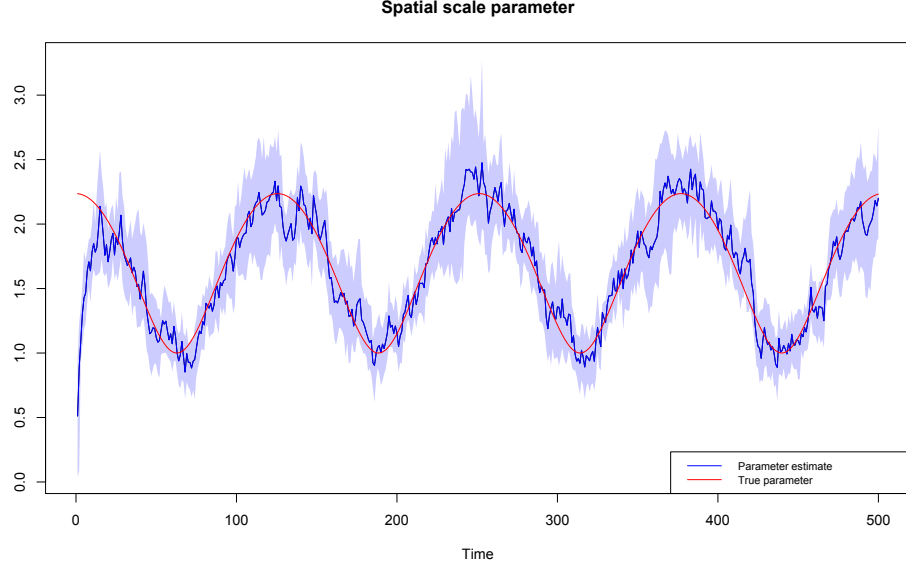


FIGURE 4.6 – Estimation of the spatial scale parameter $l_{Q,t}$ with the PF-EnKF when $p = n$ (the state is fully observed).

Focusing now on the state estimation, our algorithm is compared to the EnKF with true Q_t . This latter uses the theoretical covariance matrix of the forecasted members $P_t^f = P_t^p + Q_t$, instead of the empirical one, to reduce the impact of the sampling error as our algorithm with Eq. (4.14). For the comparison, the experiment is repeated 10 times with the same true states and observations for the two algorithms. Their global RMSE and coverage probability are computed for each experiment. The global RMSE is given by

$$\frac{1}{T} \sum_{t=1}^T \sqrt{\frac{1}{nN_{memb}} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{a,i} - \mathbf{x}_t)^\top (\mathbf{x}_t^{a,i} - \mathbf{x}_t)}.$$

The coverage probability represents the probability to have $\mathbf{x}_{t,k}$, with $k \in \{1, \dots, n\}$, falling in the 95% confidence interval obtained by the algorithm at time t . The optimal value for the coverage probability is 0.95 as explained by [46]. The global RMSE gives informations on the ensemble mean and the coverage probability on the ensemble spread. The results shown in Table 4.1 are quite similar for the two algorithms. The global RMSE and coverage probability of the PF-EnKF indicate that it generates members close to \mathbf{x}_t . The standard deviations computed for the global RMSE and coverage probability of our algorithm show its stability. Also, increasing the number of particles N_θ has not led to better results. The PF-EnKF is thus able to retrieve the true state, estimating Q_t with few extra computational cost compared to the EnKF

with true \mathbf{Q}_t . Similar results have been obtained by the PF-EnKF when \mathbf{R}_t is estimated.

Algorithm	Global RMSE	Coverage probability
PF-EnKF	1.19 \pm 0.03	0.95 \pm 0.01
EnKF with \mathbf{Q}_t	1.09 \pm 0.01	0.94 \pm 0.01

TABLE 4.1 – Comparison between the PF-EnKF with estimated \mathbf{Q}_t and the EnKF with true \mathbf{Q}_t . Each result corresponds to the average over the 10 experiments \pm the standard deviation.

4.3.2 Results for the PF-EnKF with inflation and localization

The PF-EnKF algorithm detailed in Section 4.2.2 is used with $N_\theta = 100$ and $N_{memb} = 10$ which is smaller than in Section 4.3.1 to have an important sampling error to treat. $\widehat{\mathbf{Q}}_t = \mathbf{I}$ and $\widehat{\mathbf{R}}_t = \mathbf{I}$ are respectively a priori estimates of \mathbf{Q}_t and \mathbf{R}_t that are compensated by the inflation and localization of \mathbf{P}_t^f with $\mathbf{L}(\boldsymbol{\theta}_{\mathbf{L},t})$ whose the inflation parameter $\lambda_{\mathbf{L},t}$ and localization parameter $l_{\mathbf{L},t}$ are estimated by the PF-EnKF.

At $t = 1$, for $j \in \{1, \dots, N_\theta\}$ we have $\boldsymbol{\theta}_{\mathbf{L},1}^{a,j} = (\lambda_{\mathbf{L},1}^{a,j}, l_{\mathbf{L},1}^{a,j})$ with $\lambda_{\mathbf{L},1}^{a,j}$ and $l_{\mathbf{L},1}^{a,j}$ respectively following the uniform distributions with strictly positive support $\mathcal{U}[0, 2\lambda_{\mathbf{L},0}]$ and $\mathcal{U}[0, 2l_{\mathbf{L},0}]$, where $(\lambda_{\mathbf{L},0}, l_{\mathbf{L},0}) = (1, 1)$.

For $t \geq 2$, the forecasted particles $\boldsymbol{\theta}_{\mathbf{L},t}^{f,j} = (\lambda_{\mathbf{L},t}^{f,j}, l_{\mathbf{L},t}^{f,j})$ are computed as in Section 4.3.1 using the random walks $\lambda_{\mathbf{L},t}^{f,j} = \max(\lambda_{\mathbf{L},t-1}^{a,j} + \tilde{\lambda}_{\mathbf{L},t}^j, v_{min})$ and $l_{\mathbf{L},t}^{f,j} = \max(l_{\mathbf{L},t-1}^{a,j} + \tilde{l}_{\mathbf{L},t}^j, v_{min})$ for $j \in \{1, \dots, N_\theta\}$ and $v_{min} = 10^{-4}$, with the additive noises $\tilde{\lambda}_{\mathbf{L},t}^j \sim \mathcal{N}(0, \sigma_{\lambda,\mathbf{L}}^2)$ and $\tilde{l}_{\mathbf{L},t}^j \sim \mathcal{N}(0, \sigma_{l,\mathbf{L}}^2)$. The hyperparameters are set to $\sigma_{\lambda,\mathbf{L}} = 0.1$ and $\sigma_{l,\mathbf{L}} = 1$.

The parameters of \mathbf{Q}_t are

$$\lambda_{\mathbf{Q},t} = 1 + 0.5 \sin\left(\frac{t}{10}\right),$$

$$l_{\mathbf{Q},t} = \sqrt{3 + 2 \cos\left(\frac{t}{20}\right)}.$$

The parameters of \mathbf{R}_t are

$$\lambda_{\mathbf{R},t} = 1 + 0.5 \sin\left(\frac{t}{20}\right),$$

$$l_{\mathbf{R},t} = \sqrt{1 + 0.5 \cos\left(\frac{t}{30}\right)}.$$

The goal is to retrieve \mathbf{x}_t for $t \in \{1, \dots, T\}$ and to estimate the inflation and localization parameters, not $\boldsymbol{\theta}_{\mathbf{Q},t}$ or $\boldsymbol{\theta}_{\mathbf{R},t}$. The PF-EnKF with inflation and localization is compared with the combination of usual covariance inflation and localization methods in the EnKF (which also uses $N_{memb} = 10$, $\widehat{\mathbf{Q}}_t = \mathbf{I}$ and $\widehat{\mathbf{R}}_t = \mathbf{I}$) where the time-invariant localization parameter l in $\text{GC}(l)$ is determined by grid search using the global RMSE on the true state and the time-varying inflation parameter λ_t is estimated using the innovation statistics of [16] with a temporal smoothing to obtain for $t \geq 2$, $\lambda_2 = 1$ and $v_{min} = 10^{-4}$

$$\lambda_{t+1} = \max(\rho\tilde{\lambda}_t + (1-\rho)\lambda_t, v_{min}) \text{ with } \tilde{\lambda}_t = \frac{(\mathbf{y}_t - \mathbf{H}\mathbf{x}_t^f)^\top (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t^f) - \text{Tr}(\widehat{\mathbf{R}}_t)}{\text{Tr}(\mathbf{H}\mathbf{P}_t^f \mathbf{H}^\top)}.$$

The related hyperparameter is set to $\rho = 0.05$.

The temporal evolution of the estimates of the localization and inflation parameters of the PF-EnKF and the combination of the traditional inflation and localization methods is shown in Figure 4.7. For the inflation parameter, the estimates of the PF-EnKF and usual inflation seem to be correlated with $\lambda_{\mathbf{Q},t} + \lambda_{\mathbf{R},t}$ which represents the level of noise in both the model and observation errors. The PF-EnKF seems to be more sensitive to $\lambda_{\mathbf{Q},t} + \lambda_{\mathbf{R},t}$. These two inflation methods allow to compensate the lack of knowledge on both \mathbf{Q}_t and \mathbf{R}_t . The PF-EnKF can be seen as a likelihood-based method whereas the traditional inflation using [16] is a moment-based method as explained by [46]. For the localization parameter, the estimate of the PF-EnKF varies around the optimal value $l = 1$ obtained by grid search for the usual localization. Figure 4.8a shows the values taken over time by the estimate of the localization parameter of the PF-EnKF for different ensemble sizes N_{memb} . The optimal value of the localization parameter obtained by grid search is also plotted for each ensemble size. The values of the estimate of the localization parameter for the PF-EnKF are mostly close to the optimal value and increase, as expected, when the ensemble size is larger. In the same manner, the values taken over time by the estimate of the localization parameter of the PF-EnKF are represented in Figure 4.8b for different values of the number of observed grid points p . Our localization parameter estimate is again in agreement with the optimal value obtained by grid search. Also, the more the state is observed, the lower the values taken over time by our localization parameter estimate are. The study in Appendix 4.5.3 shows through Figure 4.12a the adaption of our localization method to a time-varying value of p , allowing to handle a time-dependent observation operator \mathbf{H}_t . Moreover, Figure 4.12b shows that our localization parameter estimate also depends on the observation error. All these results are in accordance with the ones of [52] who illustrated that the optimal value of the localization parameter notably depends on the ensemble size, the observation density (given by p) and the observation error. The PF-EnKF is thus able to relevantly estimate both time-dependent inflation and localization parameters.

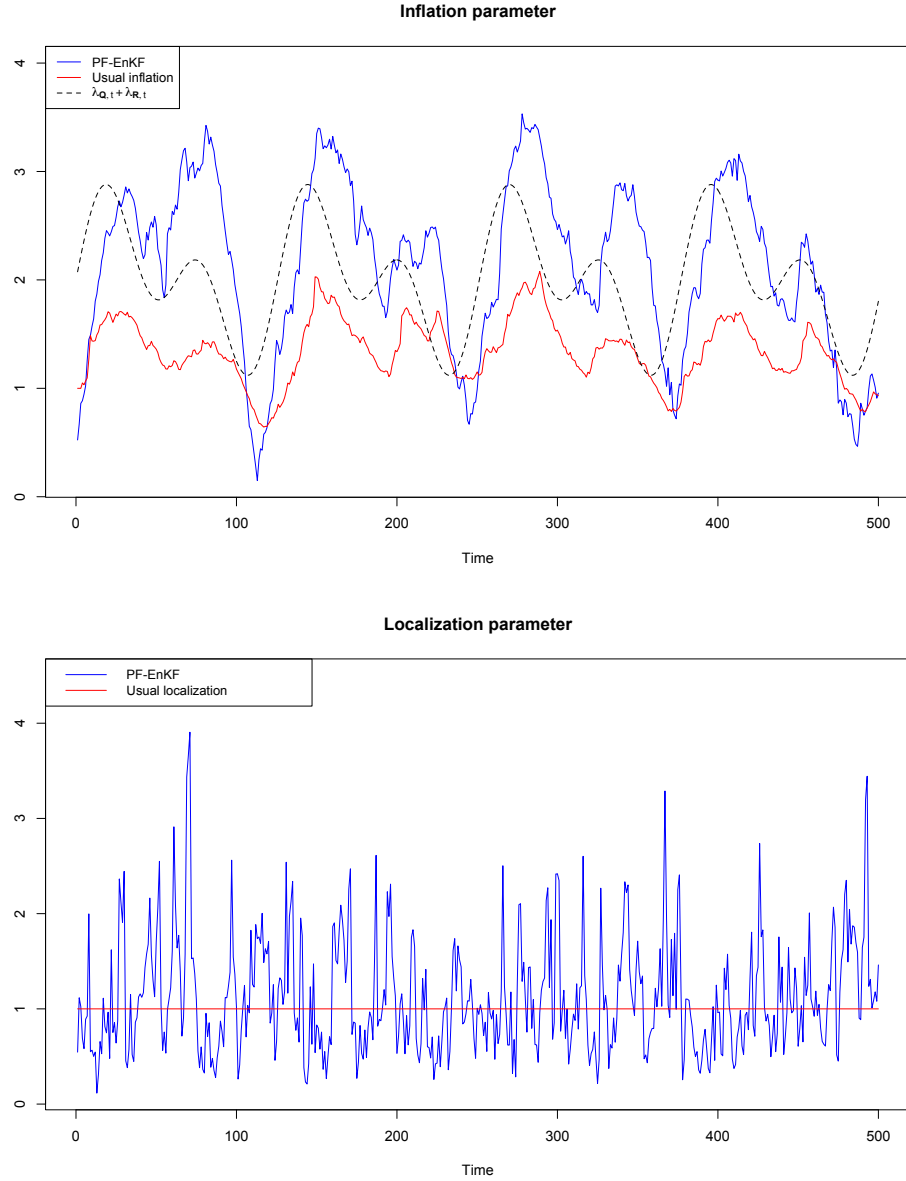
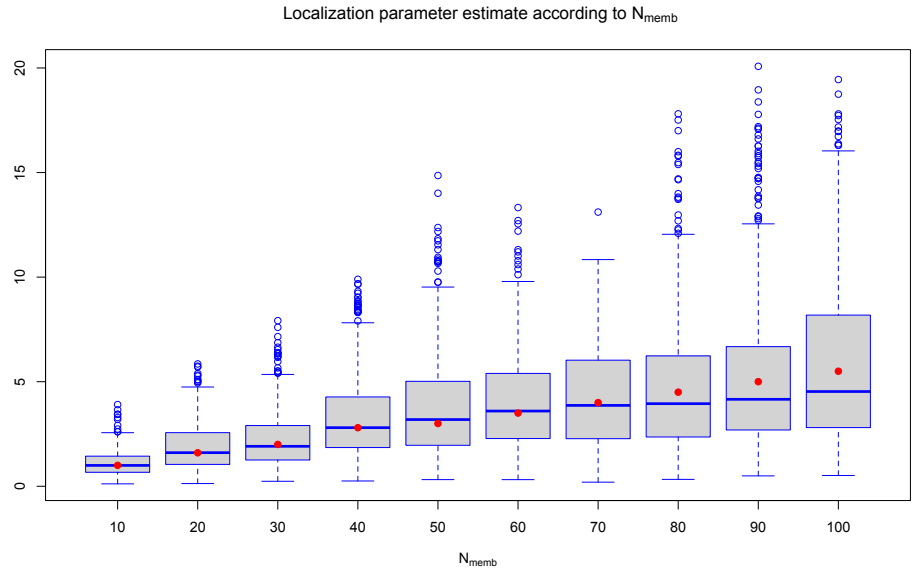
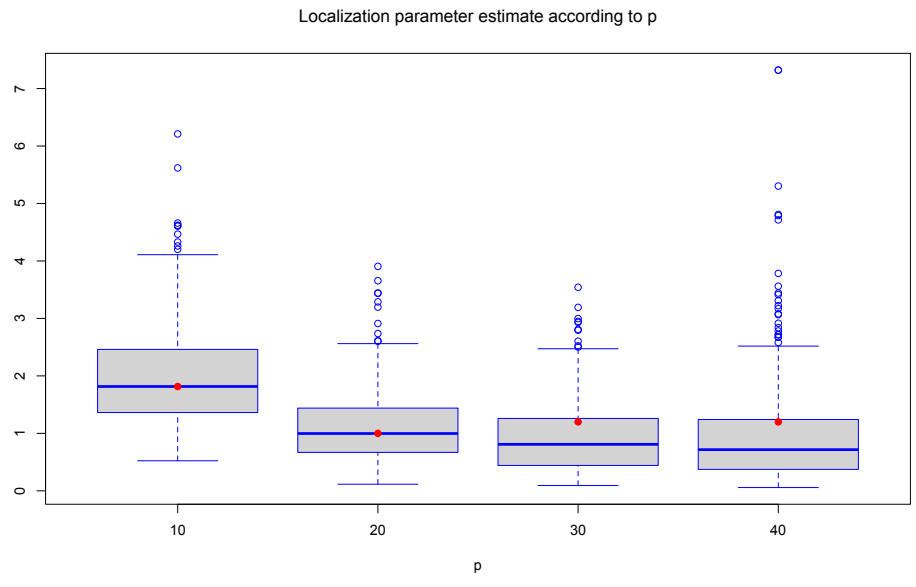


FIGURE 4.7 – Comparison of the estimates of the inflation and localization parameters of the PF-EnKF with the ones of the usual methods. For the PF-EnKF, the inflation parameter $\lambda_{L,t}$ is estimated by $\frac{1}{N_\theta} \sum_{j=1}^{N_\theta} \lambda_{L,t}^{a,j}$ and the localization parameter $l_{L,t}$ by $\frac{1}{N_\theta} \sum_{j=1}^{N_\theta} l_{L,t}^{a,j}$.



(a)



(b)

FIGURE 4.8 – Comparison of the values of the estimate of the PF-EnKF localization parameter (blue boxplot) with the optimal parameter value obtained by grid search (red point) for different ensemble sizes N_{memb} (4.8a) and for different numbers of observed grid points p when $N_{memb} = 10$ (4.8b).

For the state estimation, we compare our approach with the combination of the usual inflation and localization methods by repeating 10 times the experiment with the same true states and observations (as for Section 4.3.1). The PF-EnKF has a lower computational cost than the combination of the usual inflation and localization methods because the grid search used for the usual localization is very expensive. The global RMSE and coverage probability shown in Table 4.2 are almost the same between the two algorithms. The localization has a significant effect on the global RMSE (by reducing the sampling error impact) and the inflation on the coverage probability (by adjusting the ensemble spread). The standard deviations computed for the global RMSE and coverage probability of the PF-EnKF show, as for the usual inflation and localization, that our algorithm is stable. Good results were also obtained with a lower computational cost when a smaller number of particles $N_{\theta} = 10$ was used. The PF-EnKF with inflation and localization can thus efficiently retrieve the true state when both the model and observation errors are unknown and when a small ensemble size N_{memb} is used.

Algorithm	Global RMSE	Coverage probability
PF-EnKF	2.29 ± 0.04	0.87 ± 0.01
Usual inflation and localization	2.20 ± 0.05	0.86 ± 0.01

TABLE 4.2 – Comparison between the PF-EnKF with inflation and localization and the combination of the usual inflation and localization methods, when both \mathbf{Q}_t and \mathbf{R}_t are unknown with $N_{memb} = 10$.

4.4 Conclusion and perspectives

A new combination of the EnKF with the PF has been developed in this paper to retrieve the latent true state and to jointly estimate time-dependent stochastic parameters of the data assimilation system. To achieve this, the PF generates particles that estimate the set of stochastic parameters and the EnKF generates the related members that estimate the state and which are weighted using the PF. The generic PF-EnKF algorithm, easy to implement, has been detailed for estimating \mathbf{Q}_t , \mathbf{R}_t , or both time-dependent inflation and localization parameters. The numerical results have shown that our algorithm is able to retrieve, being stable, the true state and the parameters of \mathbf{Q}_t or \mathbf{R}_t with a reasonable computational cost. The experiments have also shown that the PF-EnKF can retrieve the true state when a small ensemble size is used and when both \mathbf{Q}_t and \mathbf{R}_t are unknown, by estimating the time-dependent inflation and localization parameters with a competitive computational cost compared to existing methods. Also, the localization parameter estimate obtained by our method has turned out to depend on the ensemble size, the number of observed grid points and the observation error.

In a future work, the PF-EnKF could be used for estimating more complex parametric forms for \mathbf{Q}_t or \mathbf{R}_t , with more parameters. Moreover it may be

applied to data-driven data assimilation, for instance when unresolved equations of the dynamical model are emulated using machine learning as in [9]. Our algorithm could also be applied to realistic problems in higher dimensions, using for example the intermediate complexity atmospheric general circulation model with simplified physical parameterizations (*SPEEDY*) detailed in [37]. Indeed, the PF-EnKF with inflation and localization may improve the performance of *SPEEDY*, which is costly, using a small ensemble size.

4.5 Appendices

4.5.1 PF-EnKF for estimating the observation error

Algorithm

The aim is to retrieve the true state \mathbf{x}_t and the set of stochastic parameters $\theta_{R,t}$ related to the observation error, assuming that \mathbf{Q}_t is known. For this purpose, the same methodology as in Section 4.2.1 is applied for the observation error where the EnKF is run with different estimates of \mathbf{R}_t . The resulting PF-EnKF method is detailed in Algorithm 7 and a cycle at time t of it is schematized in Figure 4.9.

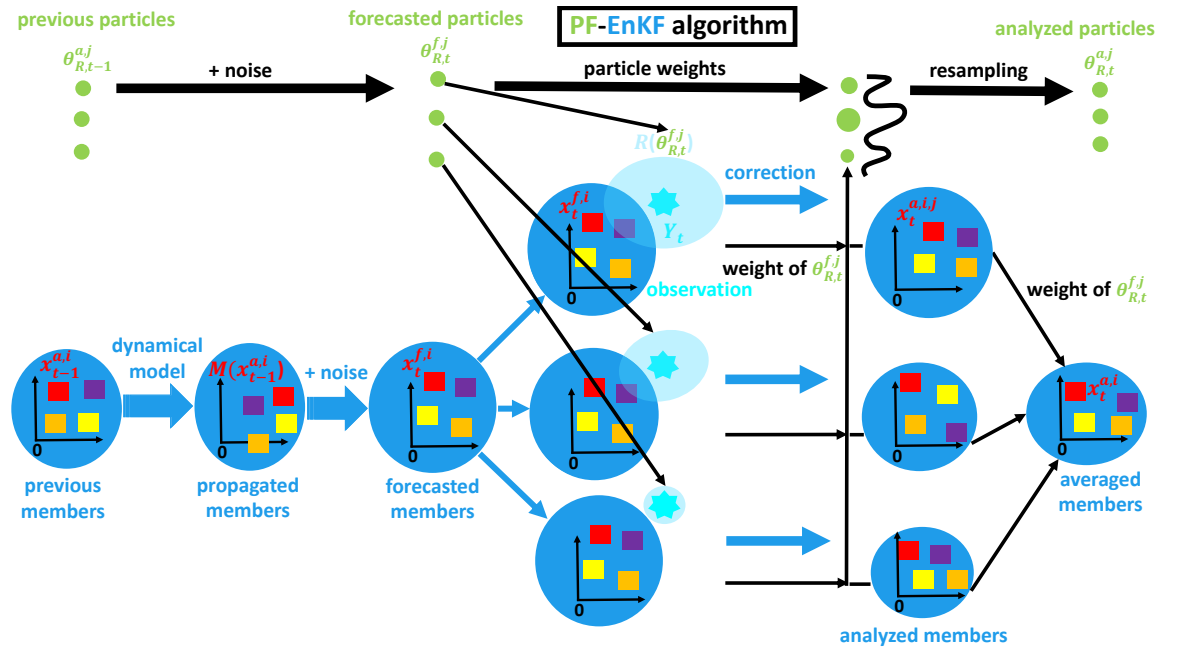


FIGURE 4.9 – PF-EnKF cycle to estimate \mathbf{R}_t .

Algorithm 7 PF-EnKF for estimating \mathbf{R}_t

Require: \mathbf{Q}_t , $\mathbf{R}(\cdot)$ and $\boldsymbol{\theta}_{\mathbf{R},0}$

- 1: *Initialization* :
- 2: **for** $i = 1, \dots, N_{memb}$ **do**
- 3: $\boldsymbol{\eta}_1^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_1)$
- 4: $\mathbf{x}_1^{a,i} = \mathbf{x}_0 + \boldsymbol{\eta}_1^i$
- 5: **end for**
- 6: **for** $j = 1, \dots, N_\theta$ **do**
- 7: $\boldsymbol{\theta}_{\mathbf{R},1}^{a,j} \sim p(\boldsymbol{\theta}_{\mathbf{R},1} | \boldsymbol{\theta}_{\mathbf{R},0})$
- 8: **end for**
- 9: **for** $t = 2, \dots, T$ **do**
- 10: *Forecast* :
- 11: **for** $i = 1, \dots, N_{memb}$ **do**
- 12: $\mathbf{x}_t^{p,i} = M(\mathbf{x}_{t-1}^{a,i})$
- 13: $\boldsymbol{\eta}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_t)$
- 14: $\mathbf{x}_t^{f,i} = \mathbf{x}_t^{p,i} + \boldsymbol{\eta}_t^i$
- 15: $\boldsymbol{\xi}_t^i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
- 16: **end for**
- 17: $\mathbf{x}_t^p = \frac{1}{N_{memb}} \sum_{i=1}^{N_{memb}} \mathbf{x}_t^{p,i}$
- 18: $\mathbf{P}_t^p = \frac{1}{N_{memb}-1} \sum_{i=1}^{N_{memb}} (\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)(\mathbf{x}_t^{p,i} - \mathbf{x}_t^p)^\top$
- 19: $\mathbf{P}_t^f = \mathbf{P}_t^p + \mathbf{Q}_t$
- 20: **for** $j = 1, \dots, N_\theta$ **do**
- 21: $\boldsymbol{\theta}_{\mathbf{R},t}^{f,j} \sim p(\boldsymbol{\theta}_{\mathbf{R},t} | \boldsymbol{\theta}_{\mathbf{R},t-1}^{a,j})$
- 22: $\mathbf{K}_t^j = \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top + \mathbf{R}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,j}))^{-1}$
- 23: Cholesky decomposition of $\mathbf{R}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,j})$ to obtain $\mathbf{C}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,j})$
- 24: *Analysis* :
- 25: **for** $i = 1, \dots, N_{memb}$ **do**
- 26: $\mathbf{d}_t^{i,j} = \mathbf{y}_t + \mathbf{C}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,j}) \boldsymbol{\xi}_t^i - \mathbf{H} \mathbf{x}_t^{f,i}$
- 27: $\mathbf{x}_t^{a,i,j} = \mathbf{x}_t^{f,i} + \mathbf{K}_t^j \mathbf{d}_t^{i,j}$
- 28: **end for**
- 29: **end for**
- 30: **for** $j = 1, \dots, N_\theta$ **do**
- 31: $\gamma_t^j = \frac{\phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^p, \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top + \mathbf{R}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,j}))}{\sum_{k=1}^{N_\theta} \phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t^p, \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top + \mathbf{R}(\boldsymbol{\theta}_{\mathbf{R},t}^{f,k}))}$
- 32: **end for**
- 33: **for** $i = 1, \dots, N_{memb}$ **do**
- 34: $\mathbf{x}_t^{a,i} = \sum_{j=1}^{N_\theta} \gamma_t^j \mathbf{x}_t^{a,i,j}$
- 35: **end for**
- 36: resample the forecasted particles $\boldsymbol{\theta}_{\mathbf{R},t}^{f,j}$ knowing the weights γ_t^j to obtain the analyzed particles $\boldsymbol{\theta}_{\mathbf{R},t}^{a,j}$
- 37: **end for**

Numerical results

Following the same study as in Section 4.3, the PF-EnKF algorithm detailed in Appendix 4.5.1 is used with $N_{memb} = 100$, $N_{\theta} = 100$ and $\mathbf{Q}_t = 0.1\mathbf{I}$ to retrieve the variance and spatial scale parameters of \mathbf{R}_t respectively given by

$$\lambda_{\mathbf{R},t} = 1 + 0.5 \sin\left(\frac{t}{20}\right),$$
$$l_{\mathbf{R},t} = \sqrt{1 + 0.5 \cos\left(\frac{t}{30}\right)}.$$

The temporal evolution of these true parameters is compared to the one of the estimates obtained by the PF-EnKF in Figure 4.10. This latter shows that our method is able to retrieve the parameters of \mathbf{R}_t .

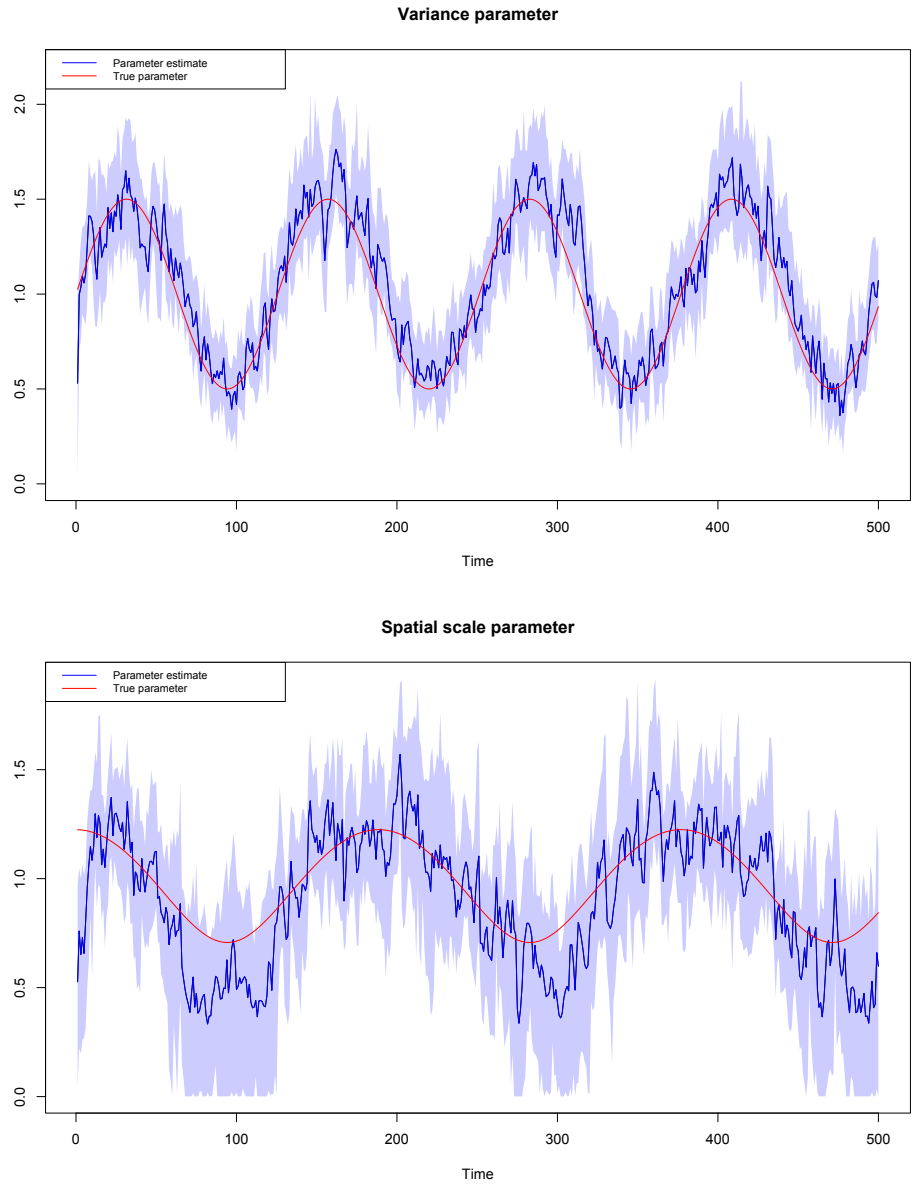


FIGURE 4.10 – Estimation of the variance parameter $\lambda_{\mathbf{R},t}$ and spatial scale parameter $l_{\mathbf{R},t}$ with the PF-EnKF algorithm (the 95% confidence interval is also plotted for each panel).

4.5.2 RMSE sensitivity to the hyperparameters $\sigma_{\lambda, Q}$ and $\sigma_{l, Q}$ of the PF-EnKF for estimating the model error

The same experiment as in Section 4.3.1 (with the same true states and observations) is repeated for different values of $\sigma_{\lambda, Q}$ first and $\sigma_{l, Q}$ then. For each hyperparameter value, the RMSE with respect to the true state is computed at every time t and all these RMSE values are represented by a boxplot in Figure 4.11. This latter shows that the RMSE is not very sensitive to the value of $\sigma_{\lambda, Q}$ and $\sigma_{l, Q}$.

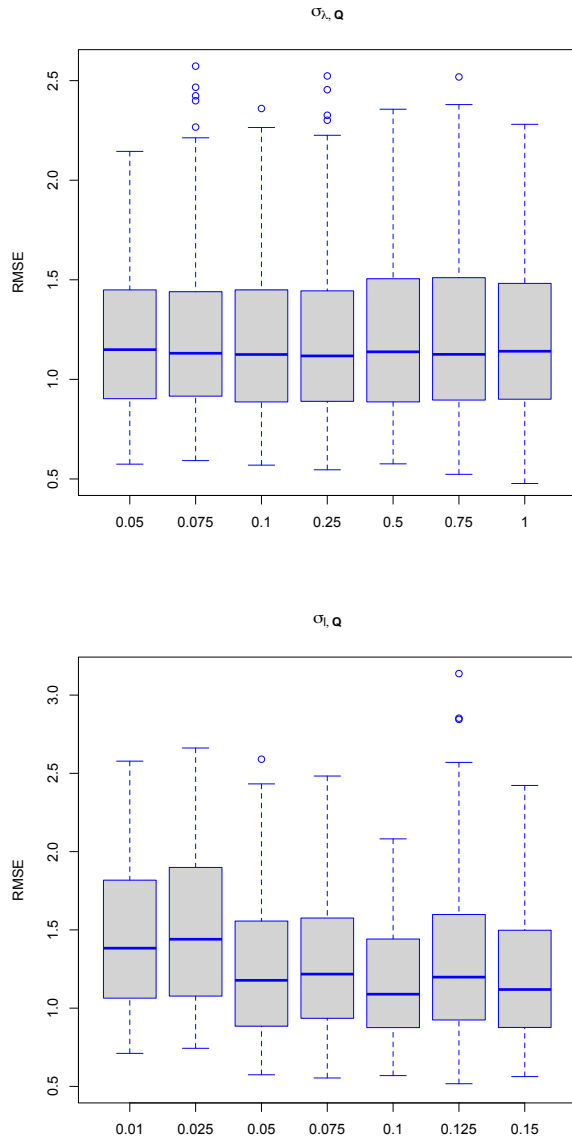


FIGURE 4.11 – RMSE values over time of the PF-EnKF when \mathbf{Q}_t is estimated, for different values of $\sigma_{\lambda, \mathbf{Q}}$ (on the top) and $\sigma_{l, \mathbf{Q}}$ (on the bottom).

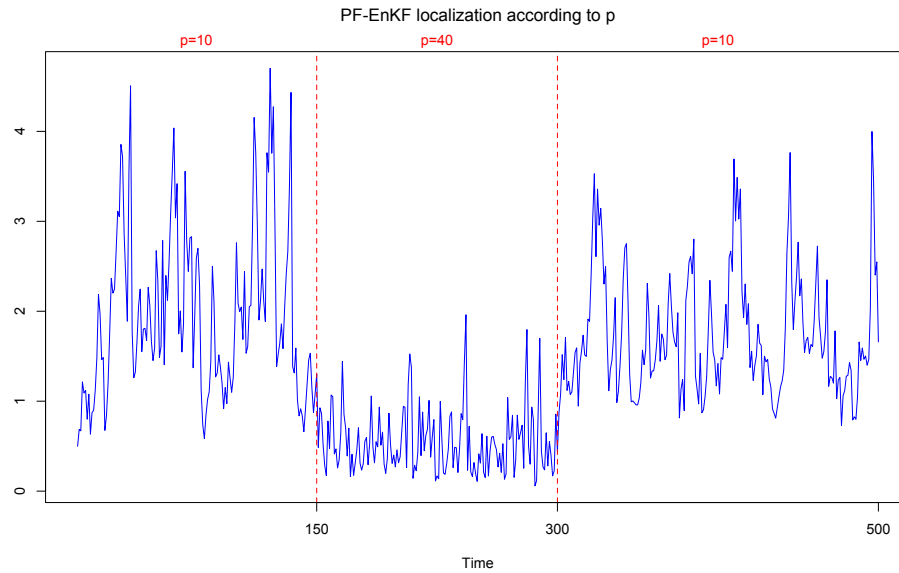
4.5.3 Sensitivity of the PF-EnKF localization to the number of observed grid points and to the observation error

Firstly, the PF-EnKF with only localization is used with the same Lorenz-96 model as in Section 4.3 and the well-known covariance matrices $\mathbf{Q}_t = \mathbf{I}$ and $\mathbf{R}_t = \mathbf{I}$. A small ensemble size $N_{memb} = 10$ is also used with $N_\theta = 100$. The temporal evolution of the estimate of the PF-EnKF localization parameter is plotted in Figure 4.12a when the number of observed grid points p varies over time. Our method adapts to the different values of p , taking lower values for the localization parameter estimate when p increases, as in Figure 4.8b.

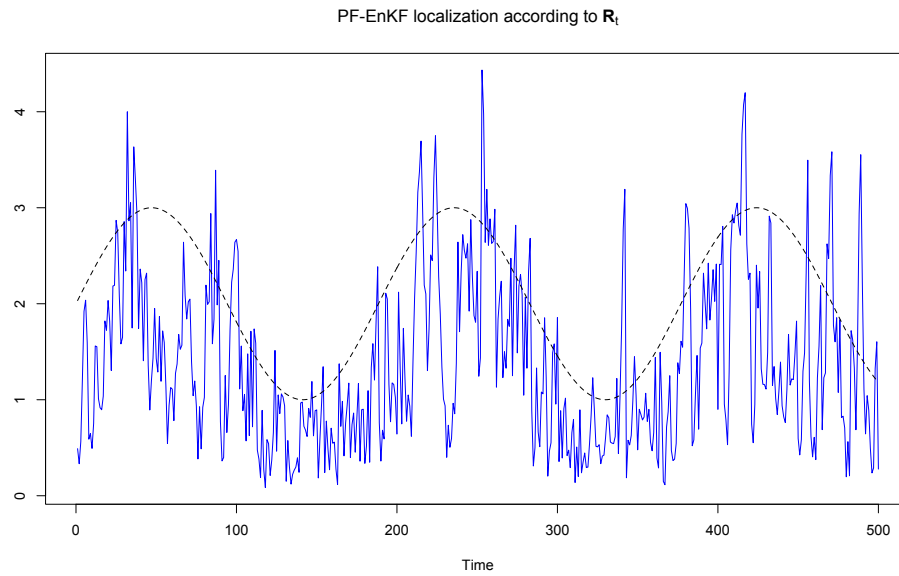
In a second part, the same experiment is led but using $p = 40$ (the state is fully observed) and the well-known covariance matrix $\mathbf{R}_t = \lambda_{\mathbf{R},t}^2 \mathbf{I}$ with $\lambda_{\mathbf{R},t} = 2 + \sin\left(\frac{t}{30}\right)$ to have a time-varying observation error. The temporal evolution of the estimate of the PF-EnKF localization parameter is compared to the one of $\lambda_{\mathbf{R},t}$ in Figure 4.12b. They seem to be correlated. The PF-EnKF localization is thus sensitive to the number of observed grid points and to the observation error.

Contributions :

- Nouvelle méthode pour estimer des paramètres variables dans le temps de \mathbf{Q}_t ou \mathbf{R}_t ,
- Nouvelle méthode pour estimer les paramètres d'inflation et de localisation.



(a)



(b)

FIGURE 4.12 – Temporal evolution of the estimate of the PF-EnKF localization parameter (blue line) according to the number of observed grid points p in 4.12a and according to the the level of noise in the observation error $\lambda_{R,t}$ (black dashed line) in 4.12b.

Chapitre 5

Conclusion générale

5.1 Bilan

Les travaux présentés dans les chapitres précédents apportent de nouvelles méthodes permettant de quantifier les incertitudes liées au modèle dynamique et aux observations en assimilation de données. Dans le Chapitre 2, une nouvelle approche a été détaillée pour davantage tenir compte de la physique du système dans la modélisation de l'erreur de modèle. Cela consiste à utiliser une solution stationnaire bruitée du vrai modèle dynamique pour modéliser l'erreur de modèle. Cette méthode, comparée à des modèles couramment utilisées pour représenter l'erreur de modèle, se révèle être particulièrement efficace lorsque la résolution temporelle des observations est faible. Dans le Chapitre 3, une nouvelle stratégie pour l'estimation en ligne de l'erreur de modèle a été étudiée. Il s'agit d'augmenter l'état avec l'erreur de modèle pour qu'elle puisse aussi être estimée par l'EnKF. Cependant, le terme aléatoire utilisé pour générer les prédictions de l'erreur de modèle doit être approprié pour guider l'EnKF dans l'apprentissage de l'erreur de modèle. La méthode alors choisie pour représenter ce terme aléatoire est celle utilisée dans le Chapitre 2, de sorte à avoir une estimation de l'erreur de modèle conforme à la physique du système. Cette méthode, comparée à une approche usuelle pour représenter le terme aléatoire, permet une estimation plus pertinente de l'erreur de modèle et de mieux retrouver l'état caché ainsi que les paramètres dynamiques inconnus. Enfin, dans le Chapitre 4, une nouvelle combinaison de l'EnKF avec le PF a été présentée pour estimer conjointement l'état caché et les paramètres des matrices de covariance associées aux erreurs de modèle (\mathbf{Q}_t) ou d'observation (\mathbf{R}_t). Pour ce faire, le PF génère des particules qui estiment le vecteur de paramètres stochastiques et l'EnKF génère les membres associés qui estiment l'état caché et sont pondérés en utilisant le PF. Les résultats numériques basés sur le modèle de Lorenz-96 ont montré que cette méthode permet de retrouver efficacement l'état caché et les paramètres variables dans le temps de \mathbf{Q}_t ou \mathbf{R}_t . L'algorithme a aussi été adapté pour estimer les paramètres d'inflation et de localisation, utiles pour

compenser la méconnaissance de \mathbf{Q}_t et \mathbf{R}_t ainsi que réduire l'impact de l'erreur d'échantillonnage sur le filtre. Les expérimentations associées ont montré la pertinence de l'estimation obtenue par notre méthode pour les paramètres d'inflation et de localisation, avec un coût algorithmique inférieur à celui de la méthode combinant les stratégies traditionnelles d'inflation et de localisation.

5.2 Perspectives

Le cas d'étude du Chapitre 2 a aussi été regardé dans le cas 2D. Cependant, pour modéliser l'erreur de modèle avec la méthode développée dans le Chapitre 2, il est nécessaire de résoudre une équation stationnaire qui n'a pas de solution analytique car il s'agit d'une équation de Poisson, ce qui ne permet pas d'utiliser directement notre méthode. Bruiter efficacement la solution de l'équation de Poisson estimée par une méthode itérative pourrait être une alternative pour représenter l'erreur de modèle et générer le terme aléatoire utilisé dans la méthode du Chapitre 3.

L'algorithme PF-EnKF détaillé dans le Chapitre 4 pourrait être utilisé pour estimer des formes paramétriques plus complexes pour \mathbf{Q}_t ou \mathbf{R}_t , avec plus de paramètres. Des essais ont été menés pour permettre au PF-EnKF d'estimer conjointement les paramètres de \mathbf{Q}_t et \mathbf{R}_t . Un résultat encourageant est montré sur la Figure 5.1, il provient de la même expérience qu'en Section 4.3. Le paramètre de variance de \mathbf{Q}_t est retrouvé correctement par l'algorithme qui arrive aussi à suivre les changements de valeur du paramètre d'échelle spatiale de \mathbf{R}_t mais avec un certain retard.

Il serait aussi intéressant de tester le PF-EnKF avec inflation et localisation détaillé en Section 4.2.2 lorsque la densité des observations évolue au cours du temps. De plus, cet algorithme pourrait servir à l'assimilation *data driven* comme [41] ou à l'assimilation *model driven* avec des modèles dynamiques coûteux nécessitant une taille d'ensemble réduite. Une stratégie pour réduire le coût de notre algorithme serait d'utiliser la moyenne des particules pour générer les analyses afin d'éviter de calculer autant de gains de Kalman qu'il y a de particules.

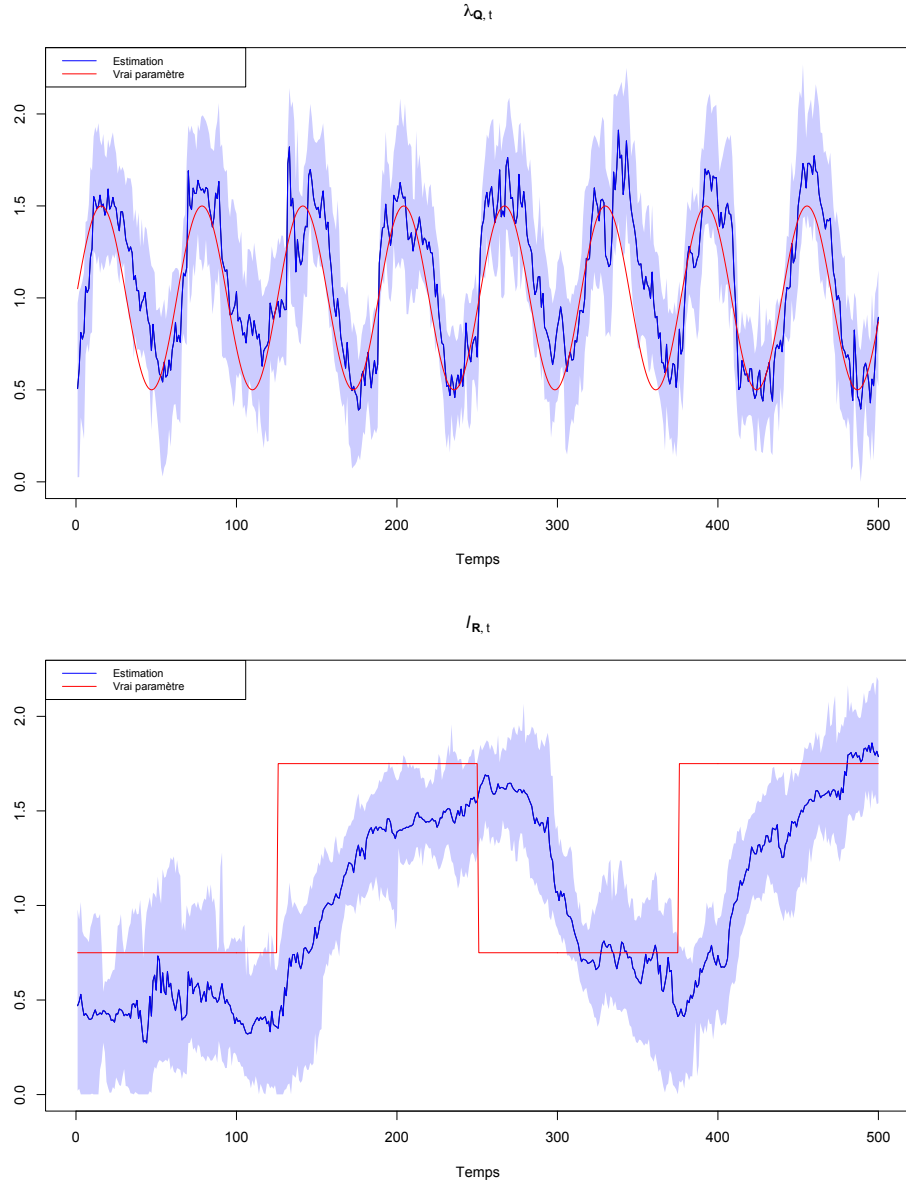


FIGURE 5.1 – Estimation conjointe des paramètres de variance de \mathbf{Q}_t et d'échelle spatiale de \mathbf{R}_t avec le PF-EnKF pour $N_{memb} = N_\theta = 100$ (l'intervalle de confiance 95% est tracé pour chaque panel). Ici $\mathbf{Q}_t = \lambda_{\mathbf{Q},t}^2 \mathbf{I}$ avec le paramètre de variance $\lambda_{\mathbf{Q},t} = 1 + 0.5 \sin\left(\frac{t}{10}\right)$ et $\mathbf{R}_t[k, k'] = e^{-\frac{d(k, k')^2}{l_{\mathbf{R},t}^2}}$ pour $(k, k') \in \{1, \dots, p\}^2$ avec le paramètre d'échelle spatiale $l_{\mathbf{R},t} = 0.75 + \mathbf{1}_{[125;250] \cup [375;T]}(t)$.

Annexe A

Équations du filtre de Kalman

On considère

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) &\sim \phi(\mathbf{x}_t; \mathbf{x}_t^f, \mathbf{P}_t^f) \\ p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) &\sim \phi(\mathbf{x}_t; \mathbf{x}_t^a, \mathbf{P}_t^a) \end{aligned}$$

avec ϕ la densité de probabilité de la distribution Gaussienne.

On s'intéresse d'abord à l'étape de prédiction :

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) \sim \phi(\mathbf{x}_t; \mathbf{x}_t^f, \mathbf{P}_t^f) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}_t^f|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)},$$

avec l'équation de Chapman-Kolmogorov on obtient

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) = \int_{\mathbb{R}^n} p(\mathbf{x}_t | \mathbf{x}_{t-1}) p(\mathbf{x}_{t-1} | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) d\mathbf{x}_{t-1}. \quad (\text{A.1})$$

On suppose que

$$\begin{aligned} p(\mathbf{x}_{t-1} | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) &\sim \phi(\mathbf{x}_{t-1}; \mathbf{x}_{t-1}^a, \mathbf{P}_{t-1}^a) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}_{t-1}^a|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a)^\top \mathbf{P}_{t-1}^{a-1} (\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a)} \\ p(\mathbf{x}_t | \mathbf{x}_{t-1}) &\sim \phi(\mathbf{x}_t; \mathbf{M}\mathbf{x}_{t-1}, \mathbf{Q}_t) = \frac{1}{(2\pi)^{n/2} |\mathbf{Q}_t|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1})^\top \mathbf{Q}_t^{-1} (\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1})}, \end{aligned}$$

en effet

$$\begin{aligned} \mathbb{E}[\mathbf{x}_t | \mathbf{x}_{t-1}] &= \mathbb{E}[\mathbf{M}\mathbf{x}_{t-1} + \boldsymbol{\eta}_t | \mathbf{x}_{t-1}] = \mathbf{M}\mathbf{x}_{t-1}, \\ \text{Cov}(\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1} | \mathbf{x}_{t-1}) &= \text{Cov}(\boldsymbol{\eta}_t | \mathbf{x}_{t-1}) = \mathbf{Q}_t. \end{aligned}$$

L'équation [\(A.1\)](#) devient donc $p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})$

$$\sim \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{n/2} |\mathbf{P}_{t-1}^a|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a)^\top \mathbf{P}_{t-1}^{a-1} (\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a)} \frac{1}{(2\pi)^{n/2} |\mathbf{Q}_t|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1})^\top \mathbf{Q}_t^{-1} (\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1})} d\mathbf{x}_{t-1} \quad (\text{A.2})$$

$$\sim \int_{\mathbb{R}^n} \frac{1}{(2\pi)^{n/2} |\mathbf{P}_{t-1}^a|^{1/2}} e^{-\frac{1}{2}\theta_1} \frac{1}{(2\pi)^{n/2} |\mathbf{Q}_t|^{1/2}} e^{-\frac{1}{2}\theta_2} d\mathbf{x}_{t-1}, \quad (\text{A.3})$$

$$\begin{aligned}
\text{avec } \theta_1 &= (\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a)^\top \mathbf{P}_{t-1}^{\alpha-1} (\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^a) \\
&= \mathbf{x}_{t-1}^\top \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1} - \mathbf{x}_{t-1}^\top \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a - \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1} + \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a, \\
\theta_2 &= (\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1})^\top \mathbf{Q}_t^{-1} (\mathbf{x}_t - \mathbf{M}\mathbf{x}_{t-1}) \\
&= \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{x}_t - \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M}\mathbf{x}_{t-1} - \mathbf{x}_{t-1}^\top \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t + \mathbf{x}_{t-1}^\top \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{M}\mathbf{x}_{t-1}.
\end{aligned}$$

En regroupant terme à terme on a

$$\begin{aligned}
\theta_1 + \theta_2 &= \mathbf{x}_{t-1}^\top (\mathbf{P}_{t-1}^{\alpha-1} + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{M}) \mathbf{x}_{t-1} - \mathbf{x}_{t-1}^\top (\mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t) - (\mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M}) \mathbf{x}_{t-1} \\
&\quad + \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{x}_t.
\end{aligned}$$

En notant $\mathbf{A}_t = \mathbf{P}_{t-1}^{\alpha-1} + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{M}$ et $\mathbf{B}_t = \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t$ on obtient

$$\begin{aligned}
\theta_1 + \theta_2 &= \mathbf{x}_{t-1}^\top \mathbf{A}_t \mathbf{x}_{t-1} - \mathbf{x}_{t-1}^\top \mathbf{B}_t - \mathbf{B}_t^\top \mathbf{x}_{t-1} + \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{x}_t \\
&= (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t)^\top \mathbf{A}_t (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t) - \underbrace{\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t + \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{x}_t}_{\theta_3}.
\end{aligned}$$

L'équation (A.3) se réécrit

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) \sim \frac{1}{(2\pi)^{n/2} |\mathbf{P}_{t-1}^\alpha|^{1/2} (2\pi)^{n/2} |\mathbf{Q}_t|^{1/2}} e^{-\frac{1}{2} \theta_3} \int_{\mathbb{R}^n} e^{-\frac{1}{2} (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t)^\top \mathbf{A}_t (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t)} d\mathbf{x}_{t-1}. \quad (\text{A.4})$$

En utilisant la propriété de la distribution Gaussienne multivariée on a

$$\int_{\mathbb{R}^n} e^{-\frac{1}{2} (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t)^\top \mathbf{A}_t (\mathbf{x}_{t-1} - \mathbf{A}_t^{-1} \mathbf{B}_t)} d\mathbf{x}_{t-1} = (2\pi)^{n/2} |\mathbf{A}_t^{-1}|^{1/2},$$

ce qui permet de simplifier (A.4)

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) \sim \frac{1}{(2\pi)^{n/2} |\mathbf{P}_{t-1}^\alpha|^{1/2} |\mathbf{Q}_t|^{1/2} |\mathbf{A}_t|^{1/2}} e^{-\frac{1}{2} \theta_3}. \quad (\text{A.5})$$

Avant de calculer θ_3 on s'intéresse à $\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t$.

$$\mathbf{B}_t^\top \mathbf{A}_t^{-1} = \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{A}_t^{-1} + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{A}_t^{-1}$$

$$\begin{aligned}
\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t &= \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{A}_t^{-1} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a + \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{A}_t^{-1} \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{A}_t^{-1} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a \\
&\quad + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{A}_t^{-1} \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t \quad (\text{A.6})
\end{aligned}$$

Le lemme d'inversion matricielle donne une formule pour \mathbf{A}_t^{-1}

$$\mathbf{A}_t^{-1} = (\mathbf{P}_{t-1}^{\alpha-1} + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{M})^{-1} = \mathbf{P}_{t-1}^\alpha - \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha,$$

qui est ensuite utilisée dans (A.6) pour obtenir

$$\begin{aligned}
\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t &= \mathbf{x}_{t-1}^{a\top} \mathbf{P}_{t-1}^{\alpha-1} \mathbf{x}_{t-1}^a - \mathbf{x}_{t-1}^{a\top} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{x}_{t-1}^a + \mathbf{x}_{t-1}^{a\top} \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t \\
&\quad - \mathbf{x}_{t-1}^{a\top} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{x}_{t-1}^a \\
&\quad - \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{x}_{t-1}^a + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t \\
&\quad - \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^\alpha \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{x}_t,
\end{aligned}$$

puis en regroupant terme à terme

$$\begin{aligned}
\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t &= \mathbf{x}_{t-1}^{\alpha \top} \mathbf{P}_{t-1}^{\alpha -1} \mathbf{x}_{t-1}^\alpha - \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{x}_{t-1}^\alpha \\
&+ \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top [\mathbf{I} - (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top] \mathbf{Q}_t^{-1} \mathbf{x}_t \\
&+ \mathbf{x}_t^\top \mathbf{Q}_t^{-1} [\mathbf{I} - \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1}] \mathbf{M} \mathbf{x}_{t-1}^\alpha \\
&+ \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top [\mathbf{I} - (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top] \mathbf{Q}_t^{-1} \mathbf{x}_t.
\end{aligned}$$

On s'intéresse maintenant au calcul de θ_3 :

$$\begin{aligned}
\theta_3 &= -\mathbf{B}_t^\top \mathbf{A}_t^{-1} \mathbf{B}_t + \mathbf{x}_{t-1}^{\alpha \top} \mathbf{P}_{t-1}^{\alpha -1} \mathbf{x}_{t-1}^\alpha + \mathbf{x}_t^\top \mathbf{Q}_t^{-1} \mathbf{x}_t = \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top \underbrace{(\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1}}_{\mathbf{F}_1} \mathbf{M} \mathbf{x}_{t-1}^\alpha \\
&- \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top \underbrace{[\mathbf{I} - (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top] \mathbf{Q}_t^{-1}}_{\mathbf{F}_2} \mathbf{x}_t \\
&- \mathbf{x}_t^\top \underbrace{\mathbf{Q}_t^{-1} [\mathbf{I} - \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1}]}_{\mathbf{F}_3} \mathbf{M} \mathbf{x}_{t-1}^\alpha \\
&+ \mathbf{x}_t^\top \underbrace{\mathbf{Q}_t^{-1} [\mathbf{I} - \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top [\mathbf{I} - (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top] \mathbf{Q}_t^{-1}]}_{\mathbf{F}_4} \mathbf{x}_t, \quad (\text{A.7})
\end{aligned}$$

en montrant que les facteurs $\mathbf{F}_1, \mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_4$ sont égaux en utilisant la formule suivante valable pour toutes matrices \mathbf{X} et \mathbf{Y} telles que \mathbf{X} et $\mathbf{X} + \mathbf{Y}$ sont inversibles

$$(\mathbf{X} + \mathbf{Y})^{-1} = [\mathbf{I} - (\mathbf{X} + \mathbf{Y})^{-1} \mathbf{Y}] \mathbf{X}^{-1}. \quad (\text{A.8})$$

Preuve.

$$\begin{aligned}
(\mathbf{X} + \mathbf{Y})^{-1} (\mathbf{X} + \mathbf{Y}) &= \mathbf{I} \\
(\mathbf{X} + \mathbf{Y})^{-1} \mathbf{X} &= \mathbf{I} - (\mathbf{X} + \mathbf{Y})^{-1} \mathbf{Y} \\
(\mathbf{X} + \mathbf{Y})^{-1} &= [\mathbf{I} - (\mathbf{X} + \mathbf{Y})^{-1} \mathbf{Y}] \mathbf{X}^{-1}
\end{aligned}$$

□

On obtient alors

$$\begin{aligned}
\mathbf{F}_1 &= (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} = [\mathbf{I} - (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top] \mathbf{Q}_t^{-1} = \mathbf{F}_2, \\
\mathbf{F}_3 &= \mathbf{F}_2^\top = \mathbf{F}_1^\top = \mathbf{F}_1, \\
\mathbf{F}_4 &= \mathbf{Q}_t^{-1} [\mathbf{I} - \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top \mathbf{F}_2] = \mathbf{Q}_t^{-1} [\mathbf{I} - \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1}] = \mathbf{F}_3.
\end{aligned}$$

En n'utilisant que \mathbf{F}_1 dans (A.7) on a

$$\begin{aligned}
\theta_3 &= \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top \mathbf{F}_1 \mathbf{M} \mathbf{x}_{t-1}^\alpha - \mathbf{x}_{t-1}^{\alpha \top} \mathbf{M}^\top \mathbf{F}_1 \mathbf{x}_t - \mathbf{x}_t^\top \mathbf{F}_1 \mathbf{M} \mathbf{x}_{t-1}^\alpha + \mathbf{x}_t^\top \mathbf{F}_1 \mathbf{x}_t \\
&= (\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^\alpha)^\top \mathbf{F}_1 (\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^\alpha) \\
&= (\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^\alpha)^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top)^{-1} (\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^\alpha).
\end{aligned}$$

Le produit des déterminants dans (A.5) peut être reformulé grâce au lemme du déterminant matriciel

$$\begin{aligned}
|\mathbf{A}_t| |\mathbf{P}_{t-1}^{\alpha}| |\mathbf{Q}_t| &= |\mathbf{P}_{t-1}^{\alpha -1} + \mathbf{M}^\top \mathbf{Q}_t^{-1} \mathbf{M}| |\mathbf{P}_{t-1}^{\alpha}| |\mathbf{Q}_t| \\
&= |\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^{\alpha} \mathbf{M}^\top|,
\end{aligned}$$

pour donner

$$\begin{aligned} p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) &\sim \frac{1}{(2\pi)^{n/2} |\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^a \mathbf{M}^\top|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^a)^\top (\mathbf{Q}_t + \mathbf{M} \mathbf{P}_{t-1}^a \mathbf{M}^\top)^{-1} (\mathbf{x}_t - \mathbf{M} \mathbf{x}_{t-1}^a)} \\ &\sim \frac{1}{(2\pi)^{n/2} |\mathbf{P}_t^f|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)}. \end{aligned}$$

On en conclut que

$$\boxed{\begin{aligned} \mathbf{x}_t^f &= \mathbf{M} \mathbf{x}_{t-1}^a, \\ \mathbf{P}_t^f &= \mathbf{M} \mathbf{P}_{t-1}^a \mathbf{M}^\top + \mathbf{Q}_t. \end{aligned}}$$

On s'intéresse maintenant à l'étape d'analyse :

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) \sim \phi(\mathbf{x}_t; \mathbf{x}_t^a, \mathbf{P}_t^a) = \frac{1}{(2\pi)^{n/2} |\mathbf{P}_t^a|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^a)^\top \mathbf{P}_t^{a-1} (\mathbf{x}_t - \mathbf{x}_t^a)},$$

grâce au théorème de Bayes on a

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})}{p(\mathbf{y}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})}. \quad (\text{A.9})$$

On suppose que

$$p(\mathbf{y}_t | \mathbf{x}_t) \sim \phi(\mathbf{y}_t; \mathbf{H} \mathbf{x}_t, \mathbf{R}_t) = \frac{1}{(2\pi)^{p/2} |\mathbf{R}_t|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)},$$

en effet

$$\begin{aligned} \mathbb{E}[\mathbf{y}_t | \mathbf{x}_t] &= \mathbb{E}[\mathbf{H} \mathbf{x}_t + \boldsymbol{\varepsilon}_t | \mathbf{x}_t] = \mathbf{H} \mathbf{x}_t, \\ \text{Cov}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t | \mathbf{x}_t) &= \text{Cov}(\boldsymbol{\varepsilon}_t | \mathbf{x}_t) = \mathbf{R}_t. \end{aligned}$$

En utilisant la formule des probabilités totales on obtient

$$p(\mathbf{y}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) = \int_{\mathbb{R}^n} p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) d\mathbf{x}_t,$$

ce qui permet de reformuler [\(A.9\)](#)

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) = \frac{p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1})}{\int_{\mathbb{R}^n} p(\mathbf{y}_t | \mathbf{x}_t) p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_{t-1}) d\mathbf{x}_t} \quad (\text{A.10})$$

$$\begin{aligned} &\sim \frac{\frac{1}{(2\pi)^{p/2} |\mathbf{R}_t|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)} \frac{1}{(2\pi)^{n/2} |\mathbf{P}_t^f|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)}}{\int_{\mathbb{R}^n} \frac{1}{(2\pi)^{p/2} |\mathbf{R}_t|^{1/2}} e^{-\frac{1}{2}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)} \frac{1}{(2\pi)^{n/2} |\mathbf{P}_t^f|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)} d\mathbf{x}_t} \quad (\text{A.11}) \end{aligned}$$

$$\begin{aligned} &\sim \frac{e^{-\frac{1}{2}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)}}{\int_{\mathbb{R}^n} e^{-\frac{1}{2}(\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t)} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f)} d\mathbf{x}_t} \quad (\text{A.12}) \end{aligned}$$

$$\sim \frac{e^{-\frac{1}{2}\beta_1(\mathbf{x}_t)} e^{-\frac{1}{2}\beta_2(\mathbf{x}_t)}}{\int_{\mathbb{R}^n} e^{-\frac{1}{2}\beta_1(\mathbf{x}_t)} e^{-\frac{1}{2}\beta_2(\mathbf{x}_t)} d\mathbf{x}_t}, \quad (\text{A.13})$$

$$\begin{aligned} \text{avec } \beta_1(\mathbf{x}_t) &= (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t)^\top \mathbf{R}_t^{-1} (\mathbf{y}_t - \mathbf{H}\mathbf{x}_t) = \mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{y}_t - \mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{H}\mathbf{x}_t - \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{x}_t^\top \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{H}\mathbf{x}_t, \\ \beta_2(\mathbf{x}_t) &= (\mathbf{x}_t - \mathbf{x}_t^f)^\top \mathbf{P}_t^{f-1} (\mathbf{x}_t - \mathbf{x}_t^f) = \mathbf{x}_t^\top \mathbf{P}_t^{f-1} \mathbf{x}_t - \mathbf{x}_t^\top \mathbf{P}_t^{f-1} \mathbf{x}_t^f - \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1} \mathbf{x}_t + \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1} \mathbf{x}_t^f. \end{aligned}$$

En regroupant terme à terme on a

$$\begin{aligned} \beta_1(\mathbf{x}_t) + \beta_2(\mathbf{x}_t) &= \mathbf{x}_t^\top (\mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{H} + \mathbf{P}_t^{f-1}) \mathbf{x}_t - \mathbf{x}_t^\top (\mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{P}_t^{f-1} \mathbf{x}_t^f) - (\mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{H} + \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1}) \mathbf{x}_t \\ &\quad + \mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1} \mathbf{x}_t^f, \end{aligned}$$

puis en notant $\mathbf{C}_t = \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{H} + \mathbf{P}_t^{f-1}$ et $\mathbf{D}_t = \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{P}_t^{f-1} \mathbf{x}_t^f$ on obtient

$$\begin{aligned} \beta_1(\mathbf{x}_t) + \beta_2(\mathbf{x}_t) &= \mathbf{x}_t^\top \mathbf{C}_t \mathbf{x}_t - \mathbf{x}_t^\top \mathbf{D}_t - \mathbf{D}_t^\top \mathbf{x}_t + \mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1} \mathbf{x}_t^f \\ &= (\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)^\top \mathbf{C}_t (\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t) - \underbrace{\mathbf{D}_t^\top \mathbf{C}_t^{-1} \mathbf{D}_t + \mathbf{y}_t^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{x}_t^f{}^\top \mathbf{P}_t^{f-1} \mathbf{x}_t^f}_{\beta_3}. \end{aligned}$$

Ce qui permet de réécrire l'équation (A.13)

$$p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) \sim \frac{e^{-\frac{1}{2}\beta_3} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)^\top \mathbf{C}_t (\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)}}{e^{-\frac{1}{2}\beta_3} \int_{\mathbb{R}^n} e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)^\top \mathbf{C}_t (\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)} d\mathbf{x}_t} \quad (\text{A.14})$$

$$\sim \frac{e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)^\top \mathbf{C}_t (\mathbf{x}_t - \mathbf{C}_t^{-1} \mathbf{D}_t)}}{(2\pi)^{n/2} |\mathbf{C}_t^{-1}|^{1/2}}. \quad (\text{A.15})$$

Le lemme d'inversion matricielle donne une expression pour \mathbf{C}_t^{-1}

$$\begin{aligned} \mathbf{C}_t^{-1} &= (\mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{H} + \mathbf{P}_t^{f-1})^{-1} \\ &= \mathbf{P}_t^f - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{P}_t^f \\ &= (\mathbf{I} - \underbrace{\mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H}}_{\text{gain } \mathbf{K}_t}) \mathbf{P}_t^f, \end{aligned}$$

ce qui permet de calculer $\mathbf{C}_t^{-1} \mathbf{D}_t$ dans (A.15)

$$\begin{aligned} \mathbf{C}_t^{-1} \mathbf{D}_t &= (\mathbf{I} - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H}) \mathbf{P}_t^f (\mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{P}_t^{f-1} \mathbf{x}_t^f) \\ &= \mathbf{P}_t^f \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t + \mathbf{x}_t^f - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top \mathbf{R}_t^{-1} \mathbf{y}_t - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{x}_t^f \\ &= \mathbf{x}_t^f + \mathbf{P}_t^f \mathbf{H}^\top [\mathbf{I} - (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top] \mathbf{R}_t^{-1} \mathbf{y}_t - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{x}_t^f, \end{aligned}$$

grâce à la formule (A.8) on a

$$[\mathbf{I} - (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top] \mathbf{R}_t^{-1} = (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1}$$

et on obtient

$$\begin{aligned} \mathbf{C}_t^{-1} \mathbf{D}_t &= \mathbf{x}_t^f + \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{y}_t - \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1} \mathbf{H} \mathbf{x}_t^f \\ &= \mathbf{x}_t^f + \underbrace{\mathbf{K}_t (\mathbf{y}_t - \mathbf{H} \mathbf{x}_t^f)}_{\text{innovation } \mathbf{d}_t}. \end{aligned}$$

L'équation (A.15) devient ainsi

$$\begin{aligned}
 p(\mathbf{x}_t | \mathbf{y}_1, \dots, \mathbf{y}_t) &\sim \frac{e^{-\frac{1}{2}(\mathbf{x}_t - (\mathbf{x}_t^f + \mathbf{K}_t \mathbf{d}_t))^\top ((\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^f)^{-1} (\mathbf{x}_t - (\mathbf{x}_t^f + \mathbf{K}_t \mathbf{d}_t))}}{(2\pi)^{n/2} |(\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^f|^{1/2}} \\
 &\sim \frac{e^{-\frac{1}{2}(\mathbf{x}_t - \mathbf{x}_t^a)^\top \mathbf{P}_t^a^{-1} (\mathbf{x}_t - \mathbf{x}_t^a)}}{(2\pi)^{n/2} |\mathbf{P}_t^a|^{1/2}},
 \end{aligned}$$

on en conclut que

$$\begin{aligned}
 \mathbf{x}_t^a &= \mathbf{x}_t^f + \mathbf{K}_t \mathbf{d}_t, \\
 \mathbf{P}_t^a &= (\mathbf{I} - \mathbf{K}_t \mathbf{H}) \mathbf{P}_t^f, \\
 \mathbf{K}_t &= \mathbf{P}_t^f \mathbf{H}^\top (\mathbf{R}_t + \mathbf{H} \mathbf{P}_t^f \mathbf{H}^\top)^{-1}, \\
 \mathbf{d}_t &= \mathbf{y}_t - \mathbf{H} \mathbf{x}_t^f.
 \end{aligned}$$

Annexe B

Partial Differential Equations for Oceanic Artificial Intelligence

Jules Guillot, Guillaume Koenig, Hadi Minbashian,
Emmanuel Frénod, H el ene Flourent and Julien Brajard

Abstract

The sea surface temperature (SST) plays a significant role in analyzing and assessing the dynamics of weather and also biological systems. It has various applications such as weather forecasting or planning of coastal activities. On the one hand, standard physical methods for forecasting SST use coupled ocean-atmosphere prediction systems, based on the Navier-Stokes equations. These models rely on multiple physical hypotheses and do not optimally exploit the informations available in the data. On the other hand, despite the availability of large amounts of data, direct applications of machine learning methods do not always lead to competitive state of the art results. Another approach is to combine these two methods : this is data-model coupling. The aim of this paper is to use a model in another domain. This model is based on a data-model coupling approach to simulate and predict SST. We first introduce the original model. Then, the modified model is described, to finish with some numerical results.

B.1 Introduction

The general task of this assignment could have been guessed from the title "Partial differential equations for oceanic artificial intelligence". The idea was to use artificial intelligence tools to determine the SST of a given oceanic zone.

This has already been done in [14]. What they exactly did is using a convolutional neural network (CNN) and a transport equation to predict the evolution of the field of surface temperature for a few days. The CNN was used to identify and predict the velocity field used in the transport equation. They did not consider the boundary conditions for the transport equation nor the diffusion coefficient.

However, other approaches can be followed, having a more physical representation of the problem by adding some terms. Machine learning has been used in this context by [53] and could identify the non-linear terms present in the output of an hydrodynamic model. In this case, they did not use a neural network, but sophisticated parameters identification tools such as stochastic gradient descent and LASSO objective function.

Another interesting approach was the one of [12] and [42]. Their idea was to mimick the behavior of a neural network using differential equations, ordinary (ODE) in the first paper and partial (PDE) in the second one. The interest of this approach would be to use the computationally efficient tools that exist for identifying parameters in differential equations to replace the backpropagation in neural networks.

In a data-model coupling context, learning PDEs (related to unknown phenomena) from data, has been done in [35]. More recently, a new neural network architecture based on PDEs has been made by [39]. Another work is currently led on the link between discretized PDEs and the layers of a deep neural network.

We decided to follow a different approach with all this in mind. We used a model with an advection-diffusion problem and some ODEs, that were already developed in [21], to fit this to our SST evolution problem. The idea was that the PDE was close enough to the physics of our problem to be efficient, and that the ODEs could be fitted in order to represent the phenomenon not taken into account in the PDE.

B.2 The model

The model used is a modified version of the model of [21]. The idea is to use an advection-diffusion equation for a given variable and have it passed through several transformations given by ODEs to account for other forcings. In our case, we are interested in the temporal evolution of the SST and our forcing variable is the solar radiation at the top of the atmosphere. To simplify our problem, we reduce it to the determination of the average of the temperature on a given zone for an entire year.

$$\frac{\partial \Phi_f}{\partial t}(t, x) + \omega \frac{\partial \Phi_f}{\partial x}(t, x) - c \frac{\partial \left[\chi \frac{\partial \Phi_f}{\partial x} \right]}{\partial x}(t, x) = Q(t, x) - fF(x)\Phi_f(t, x) \quad (\text{B.1})$$

$$\frac{\partial \Psi}{\partial t}(t, x) = fF(x)\Phi_f(t, x) - u\Psi(t, x) \quad (\text{B.2})$$

$$\frac{\partial \Xi}{\partial t}(t, x) = -u\Psi(t, x) \left(\Xi(t, x) - sup \right) - P \left(\Xi(t, x) - inf \right) \quad (\text{B.3})$$

$$s(t) = \int_{\Omega} \Xi(t, x) dx \quad (\text{B.4})$$

Here the parameters are the following : Φ_f is the unknown advected quantity, ω is the advection velocity, c and χ are respectively the scaling of the diffusion and the field of the coefficient of diffusion. Q corresponds to the source term and is here given by a scaled version of the solar radiation at the top of the atmosphere. f and F are terms transferring quantities from the advection-diffusion equation towards the ODEs. This was inspired from a biological model where this transfer consists in the extraction of nutrients from the flow by biological agents.

The parameter Ψ is the first "hidden quantity" related to the non-represented phenomenon. We see that it increases by extracting quantity from the PDE and decreases with the term $u\Psi$ which transfers some of its content to the next ODE.

The quantity Ξ corresponds to another hidden variable. Its behavior is slightly different from Ψ because its equation is not linear and contains saturation terms. The first term links the evolution of Ξ to what is extracted from the quantity Ψ .

Finally we have the quantity $s(t)$ which basically contains the total of the quantity Ξ at each time t .

In our case, we thought that the PDE was able to correctly reproduce the heat absorbed by the water when the solar radiation arrives. And we thought that the ODEs would be able to reproduce efficiently other phenomena affecting the link between incoming solar radiation and water surface temperature, such as the cloud coverage, the difference in opacity of water or the wind chill for example. But if we wanted those phenomena to be correctly reproduced we needed to change some of the parameters.

B.3 Application to our problem

B.3.1 General view of the problem

As said earlier, we tried to identify a working relationship between solar radiation at the top of the atmosphere and mean value of the SST. To do so, we tried to fix three parameters in our system : *sup*, *inf* and u . Those are not related to the PDE but to the ODEs. We have used data taken from some part of the North Atlantic in Figure [B.1](#) and solar radiation that we collected from the [CERES project](#).

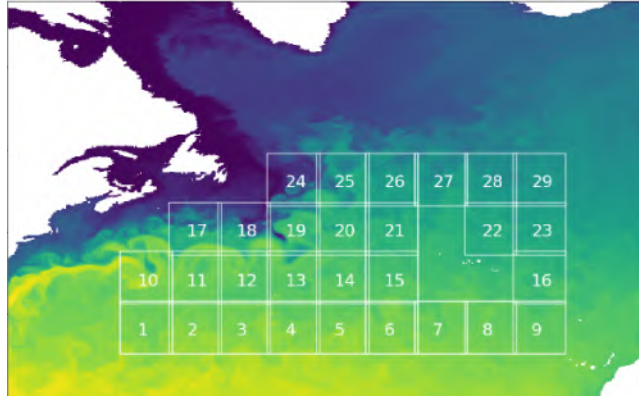


FIGURE B.1 – Localisation of our dataset. It is present in the rectangle 20.

Finally, we got interested in the spatial average of those data. Also, the variations over the years were of interest because they would give us a minimum error to reach with our model. We then did the spatial averaging which is presented in Figure [B.2](#).

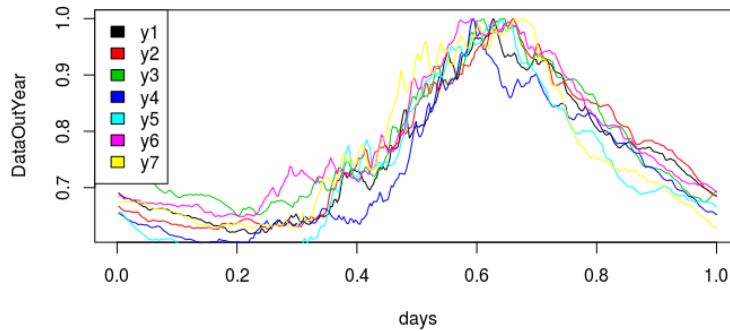


FIGURE B.2 – Variations of the spatial average of the temperature over different years. This has been normalized to be compatible with our model.

B.3.2 More details on the data

Looking at the data, we see a very clear seasonal trend associated with some high frequency variations. This seasonal trend is common to every year while the high frequency variations differ from one year to another. At first we will try to identify this seasonal trend, because it is directly linked to the sun radiations arriving on the surface. To see this, we can have a look at Figure [B.3](#)

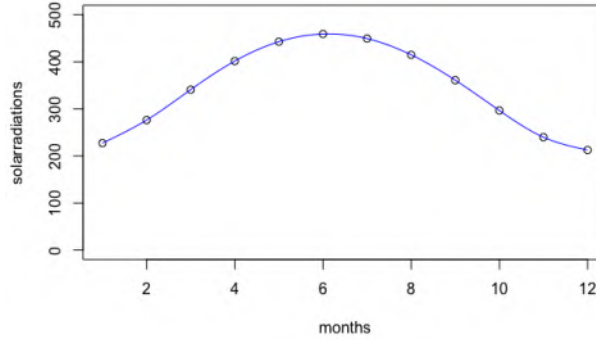


FIGURE B.3 – Sun radiation arriving at the top of the atmosphere for a standard year in $W.m^{-2}$. The variations between different years are very weak and therefore this year is taken as representative.

B.3.3 Physics behind the model

Though we see a global connection in the shape of the curves of SST and solar radiations, we also see discrepancies. The two large discrepancies are the delay between the moment of highest irradiance (in June) and the moment of highest temperature (in July and August) and the high frequency signal in temperature, which is absent from the solar radiation signal. Finally, the important interannual variations in temperature that are not present in solar radiation are also intriguing.

It is interesting to consider the origin of those data before trying to explain the discrepancies between the curves. First of all, the temperature data come from a model of oceanic circulation (NEMO) and are taken from the upper region of the oceanic model. This upper region is taken as the highest region in the vertical discretization of the model. This highest region is then a cell that represents a few meters of water. Both physically and numerically, this region is affected by complex phenomena that we will describe further in this paper. It is also interesting to note that this model is corrected by a variational algorithm that incorporates satellite data for the SST. However, those temperatures correspond to the skin temperature of the water, which is a layer of a few centimeters between the atmosphere and the upper ocean. It is often considered that the temperature in this layer is correlated with the temperature of the first meters of water.

The data for solar radiations come from satellite observations given at the CERES. They correspond to the solar radiations reaching the top of the atmosphere, and not the solar radiations reaching the surface of the Earth. Therefore, they are not affected by meteorological phenomena such as cloud coverage or the concentration of chemical species in the atmosphere. That explains the small interannual variability. Therefore, the major part of their variations is due to the tilt of the rotation axis of the Earth and to the variation of the distance Earth-Sun during a year. This distance and the tilt of the rotation axis of Earth change during each year, but very slowly. That explains the low interannual variability and the absence of high frequency signal.

Now we can try to point out some phenomena that explain interannual variability

and the high frequency signal in the SST. As stated above, the state of the atmosphere, such as cloud coverage or the concentration of some chemical species in the atmosphere can have a strong effect on the solar radiations reaching the surface and therefore on the energy available to heat the ocean. Purely oceanic parameters, such as the state of the sea surface, the opacity of water or the depth of the surface layer may also affect the way the radiation heat up the upper layer. Finally, external phenomena such as the temperature of the water when it enters the zone, the temperature of the atmosphere or the heat transfer with the lower part of the ocean may also play a role. All those phenomena may affect both the high frequency variations and the delay observed, as well as the interannual variability of the SST.

Another effect that plays a role in the delay is the high heat capacity of water. Therefore, it takes some time before the sun radiations heat the water up and we end up with a maximum of temperature that is delayed compared with the maximum of solar radiations.

B.3.4 Link with the model

In our case, we use the solar radiation forcing directly into the advection-diffusion equation. We therefore count on it to transfer the part directly linked to the solar radiation. However, to model all the other phenomena responsible notably for the delay or the high frequencies, we rely on the ODEs. For this reason, we decided to focus our efforts on the determination of the parameters of the ODEs.

B.4 Numerical results

We begin by testing the model of [21] directly with our forcings and our SST data and try to see if it is able to reproduce a correct behavior. We first learn the different *sup*, *inf* and *u* parameters for an average of the different years for the temperature. The results are in Figure B.4 and obtained with the input shown in Figure B.5. The parameters of the model are presented in Table B.1.

TABLE B.1 – Values of the parameters.

Parameter	Value
ω	20
c	0.0002
f	500
P	4.3
u	150
<i>sup</i>	13.5
<i>inf</i>	0.44

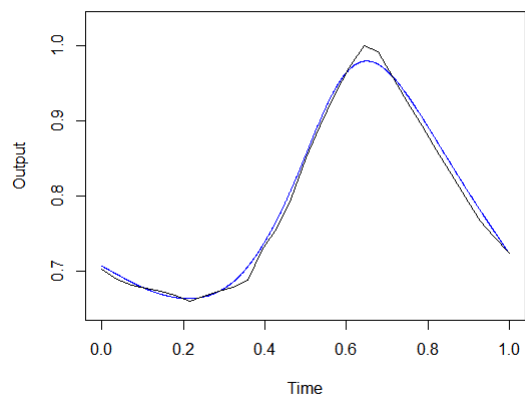


FIGURE B.4 – Comparison between the model outputs (in blue) and the data for an average year (in black).

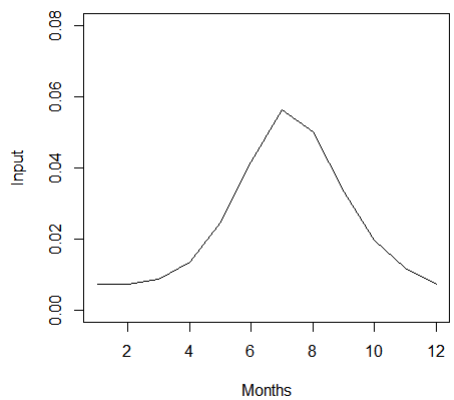


FIGURE B.5 – Input function used for the model of [\[21\]](#).

B.4.1 Interannual variability

We then try the same methodology but for the SST of different years. Since the solar radiations vary only slightly, we keep the same values for each year. The idea is to see how the model adapts to those variations. The data are presented in Figure [B.6](#) and the associated parameters in Table [B.2](#).

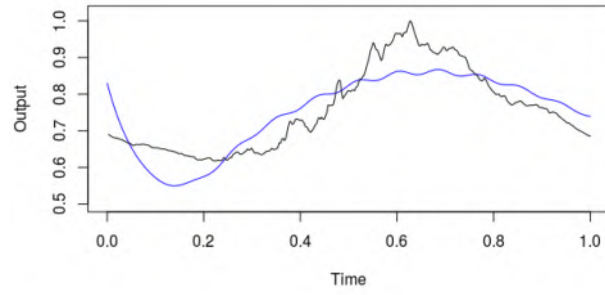
Year	<i>sup</i>	<i>inf</i>	<i>u</i>	objective function
1	56.6872427983148	0.5037952	199.999999999961	0.0488598607471862
2	68.4254272816738	0.4976314	199.999993030828	0.0461400247339915
3	51.7736625511663	0.5224552	199.999999999882	0.0579528378552093
4	62.5269300303108	0.4888889	199.999999999908	0.0554632691635029
5	68.8096166836958	0.4444428	199.999811832358	0.0640272984624972
6	52.5925925925795	0.5375095	199.999999999961	0.0575910849643628
7	54.1394604472804	0.5199167	199.999999999961	0.0733631601644184

TABLE B.2 – Parameters determined for the different years.

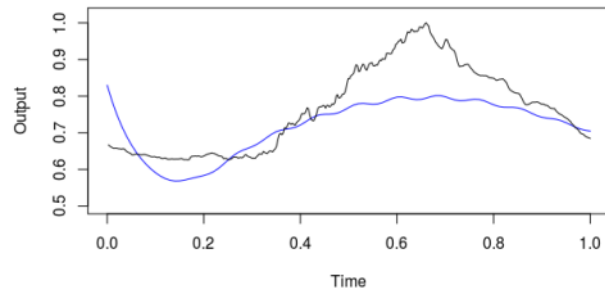
There are some obvious features. We see that there is a correct fitting in the timing of the highest temperature, but that the values are quite off. The period of spring though is correctly fitted. Maybe the two biggest discrepancies are the beginning of the year and the fact that there is no high frequency variations. For the high frequency, since we have a low frequency input function and a linear model applied to it, it is expected.

B.4.2 2D-extension of the model

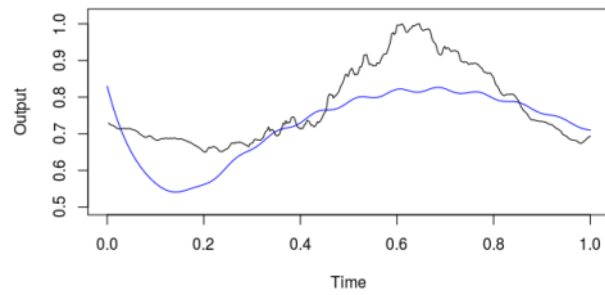
We also tried to use *Freefem++* and *Fenics* to build a 2D representation of the model. Both are finite element modelling softwares. We used them for solving the PDE and the explicit Euler scheme for the ODEs. However, we were facing an issue : the finite element method is not suitable for an advection-diffusion problem. Therefore, we decided to only use the advection problem and to take advantage of the numerical diffusion to play the role of the physical diffusion. This approach was limited since it is really dependent on the scale of the elements and on the time step used. Also, it is quite sensitive to the family of elements used. A first example of discretization and solving of the advection problem is shown in Figure [B.7](#). We also tried it with a finer discretization as shown in Figure [B.8](#). We observed less diffusion as expected and still were using a second order polynomial basis. Another tryout was to see the impact of the time step on the diffusion as displayed in Figure [B.9](#). Finally we tried with a discontinuous Galerkin basis to diminish the diffusion in Figure [B.10](#).



(a) Year 1

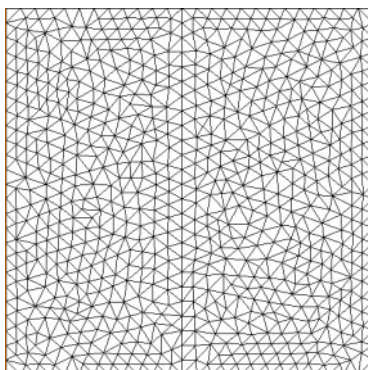


(b) Year 2

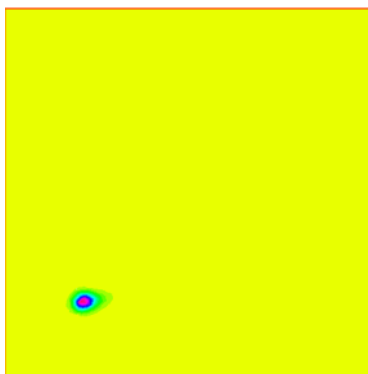


(c) Year 3

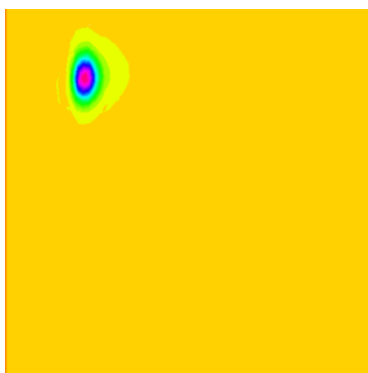
FIGURE B.6 – Comparison between the model outputs (in blue) and the data (in black) for different years.



(a)

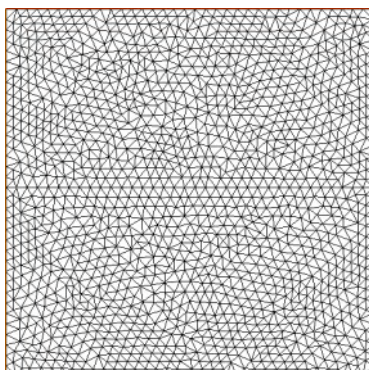


(b)

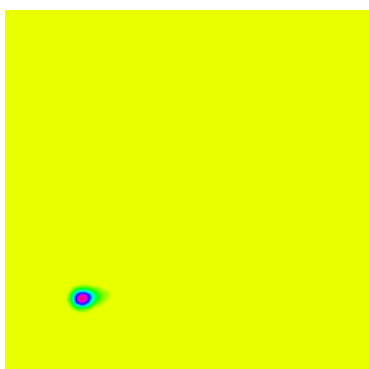


(c)

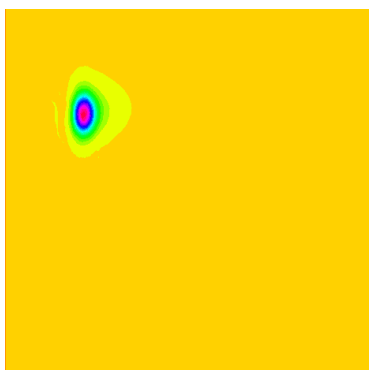
FIGURE B.7 – Grid (B.7a), initial condition (B.7b) and result of the advection model after a few time step (B.7c) using *Freefem++* with only an advection term and the ODEs.



(a)



(b)



(c)

FIGURE B.8 – Finer grid (B.8a), initial condition (B.8b) and result after a few time step (B.8c) for the advection problem and ODEs.

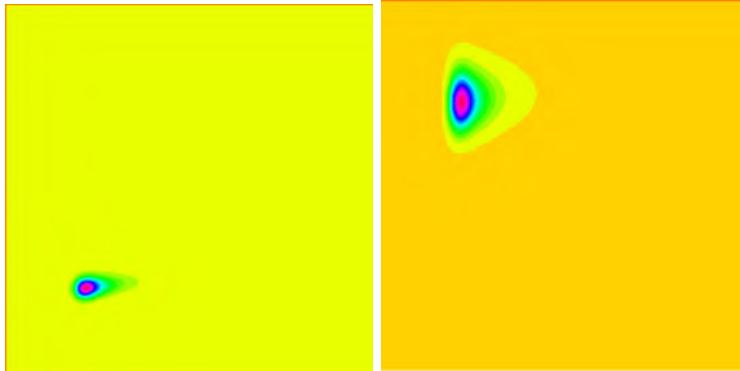


FIGURE B.9 – Same situation as in Figure [B.8](#) but with a different time step.

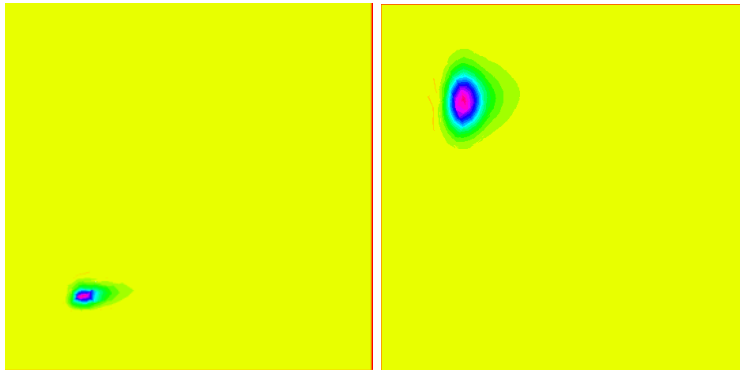


FIGURE B.10 – Same situation as in Figure [B.9](#) but with a discontinuous Galerkin basis.

B.5 Conclusion

We have adapted a model, using a data-model coupling approach, to another framework with the same underlying phenomena : the SST estimation. This model contains a PDE and ODEs with parameters learnt thanks to the available data. To go further, we could use a neural network to optimize those parameters instead of the statistical learning tool of [\[21\]](#).

Bibliographie

- [1] Pierre Ailliot, Anne Cuzol, Gilles Durrieu, H el ene Flourent, Emmanuel Fr enod, Jules Guillot, Jean-Paul Lucas, and Fran ois Septier. Synth ese des questions math ematiques soulev ees par la mise en oeuvre de jumeaux num eriques pour le suivi et le pilotage de syst emes dynamiques en entreprises. *hal-03167416*, 2021.
- [2] Boujemaa Ait-El-Fquih and Ibrahim Hoteit. A particle-filter based adaptive inflation scheme for the ensemble kalman filter. *Quarterly Journal of the Royal Meteorological Society*, 146(727) :922–937, 2020.
- [3] Jeffrey L Anderson. An ensemble adjustment kalman filter for data assimilation. *Monthly weather review*, 129(12) :2884–2903, 2001.
- [4] Jeffrey L Anderson. An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus A : Dynamic meteorology and oceanography*, 59(2) :210–224, 2007.
- [5] Jeffrey L Anderson. Localization and sampling error correction in ensemble kalman filter data assimilation. *Monthly Weather Review*, 140(7) :2359–2371, 2012.
- [6] Jeffrey L Anderson and Stephen L Anderson. A monte carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly weather review*, 127(12) :2741–2758, 1999.
- [7] Ross N Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703) :607–633, 2017.
- [8] Judith Berner, Ulrich Achatz, Lauriane Batte, Lisa Bengtsson, Alvaro De La Camara, Hannah M Christensen, Matteo Colangeli, Danielle RB Coleman, Daan Crommelin, Stamen I Dolaptchiev, et al. Stochastic parameterization : Toward a new view of weather and climate models. *Bulletin of the American Meteorological Society*, 98(3) :565–588, 2017.
- [9] Julien Brajard, Alberto Carrassi, Marc Bocquet, and Laurent Bertino. Combining data assimilation and machine learning to infer unresolved scale parametrization. *Philosophical Transactions of the Royal Society A*, 379(2194) :20200086, 2021.
- [10] Gerrit Burgers, Peter Jan Van Leeuwen, and Geir Evensen. Analysis scheme in the ensemble kalman filter. *Monthly weather review*, 126(6) :1719–1724, 1998.

- [11] Alberto Carrassi, Marc Bocquet, Laurent Bertino, and Geir Evensen. Data assimilation in the geosciences : An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews : Climate Change*, 9(5) :e535, 2018.
- [12] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.
- [13] Tadeo J Cocucci, Manuel Pulido, Magdalena Lucini, and Pierre Tandeo. Model error covariance estimation in particle and ensemble kalman filters using an online expectation–maximization algorithm. *Quarterly Journal of the Royal Meteorological Society*, 147(734) :526–543, 2021.
- [14] Emmanuel De Bézenac, Arthur Pajot, and Patrick Gallinari. Deep learning for physical processes : Incorporating prior scientific knowledge. *Journal of Statistical Mechanics : Theory and Experiment*, 2019(12) :124009, 2019.
- [15] Timothy DelSole and Xiaosong Yang. State and parameter estimation in stochastic dynamical models. *Physica D : Nonlinear Phenomena*, 239(18) :1781–1788, 2010.
- [16] Gérald Desroziers, Loic Berre, Bernard Chapnik, and Paul Poli. Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society : A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613) :3385–3396, 2005.
- [17] Geir Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research : Oceans*, 99(C5) :10143–10162, 1994.
- [18] Geir Evensen and Peter Jan Van Leeuwen. An ensemble kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6) :1852–1867, 2000.
- [19] Alban Farchi. *On the localisation of ensemble data assimilation methods*. PhD thesis, Paris Est, 2019.
- [20] Alban Farchi, Patrick Laloyaux, Massimo Bonavita, and Marc Bocquet. Using machine learning to correct model error in data assimilation and forecast applications. *Quarterly Journal of the Royal Meteorological Society*, 147(739) :3067–3084, 2021.
- [21] Hélène Flourent, Emmanuel Frénod, and Vincent Sincholle. An innovating statistical learning tool based on partial differential equations, intending livestock data assimilation. *arXiv preprint arXiv :1903.12423*, 2019.
- [22] Marco Frei and Hans R Künsch. Sequential state and observation noise covariance estimation using combined ensemble kalman and particle filters. *Monthly Weather Review*, 140(5) :1476–1495, 2012.
- [23] Gregory Gaspari and Stephen E Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554) :723–757, 1999.

- [24] Michael Ghil and Paola Malanotte-Rizzoli. Data assimilation in meteorology and oceanography. *Advances in geophysics*, 33 :141–266, 1991.
- [25] Neil J Gordon, David J Salmond, and Adrian FM Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. In *IEE proceedings F (radar and signal processing)*, volume 140, pages 107–113. IET, 1993.
- [26] Jules Guillot, Emmanuel Frénod, and Pierre Ailliot. Physics informed model error for data assimilation. *Discrete and Continuous Dynamical Systems-Series S*, 2022.
- [27] John Harlim. Model error in data assimilation. *arXiv preprint arXiv :1311.3579*, 2013.
- [28] Harrie-Jan Hendricks Franssen and W Kinzelbach. Real-time groundwater flow modeling with the ensemble kalman filter : Joint estimation of states and parameters and the filter inbreeding problem. *Water Resources Research*, 44(9), 2008.
- [29] Peter L Houtekamer and Herschel L Mitchell. A sequential ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1) :123–137, 2001.
- [30] Peter L Houtekamer and Fuqing Zhang. Review of the ensemble kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 144(12) :4489–4532, 2016.
- [31] Tijana Janjić, Niels Bormann, Marc Bocquet, JA Carton, SE Cohn, Sarah L Dance, SN Losa, Nancy K Nichols, R Potthast, Joanne A Waller, et al. On the representation error in data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 144(713) :1257–1278, 2018.
- [32] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.
- [33] François Le Gland, Valérie Monbet, and Vu-Duc Tran. *Large sample asymptotics for the ensemble Kalman filter*. PhD thesis, INRIA, 2009.
- [34] Redouane Lguensat, Pierre Tandeo, Pierre Ailliot, Manuel Pulido, and Ronan Fablet. The analog data assimilation. *Monthly Weather Review*, 145(10) :4093–4107, 2017.
- [35] Zichao Long, Yiping Lu, Xianzhong Ma, and Bin Dong. Pde-net : Learning pdes from data. In *International conference on machine learning*, pages 3208–3216. PMLR, 2018.
- [36] Edward N Lorenz. Predictability : A problem partly solved. In *Proc. Seminar on predictability*, volume 1, 1996.
- [37] Franco Molteni. Atmospheric simulations using a gcm with simplified physical parametrizations. i : Model climatology and variability in multi-decadal experiments. *Climate Dynamics*, 20, 2003.

- [38] Tim N Palmer, Roberto Buizza, F Doblas-Reyes, Thomas Jung, Martin Leutbecher, Glenn J Shutts, Martin Steinheimer, and Antje Weisheimer. Stochastic parametrization and model uncertainty. *ECMWF Technical Memoranda*, 2009.
- [39] Olivier Pannekoucke and Ronan Fablet. Pde-netgen 1.0 : from symbolic partial differential equation (pde) representations of physical processes to trainable neural network representations. *Geoscientific Model Development*, 13(7) :3373–3382, 2020.
- [40] Donald B Rubin. Using the sir algorithm to simulate posterior distributions. *Bayesian statistics*, 3 :395–402, 1988.
- [41] Juan Ruiz, Pierre Ailliot, Thi Tuyet Trang Chau, Pierre Le Bras, Valérie Monbet, Florian Sévellec, and Pierre Tandeo. Analog data assimilation for the selection of suitable general circulation models. *Geoscientific Model Development*, 15(18) :7203–7220, 2022.
- [42] Lars Ruthotto and Eldad Haber. Deep neural networks motivated by partial differential equations. *Journal of Mathematical Imaging and Vision*, 62 :352–364, 2020.
- [43] Guillermo Scheffler, Juan Ruiz, and Manuel Pulido. Inference of stochastic parametrizations for model error treatment using nested ensemble kalman filters. *Quarterly Journal of the Royal Meteorological Society*, 145(722) :2028–2045, 2019.
- [44] Chris Snyder, Thomas Bengtsson, Peter Bickel, and Jeff Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12) :4629–4640, 2008.
- [45] Jonathan R Stroud, Matthias Katzfuss, and Christopher K Wikle. A bayesian adaptive ensemble kalman filter for sequential state and parameter estimation. *Monthly weather review*, 146(1) :373–386, 2018.
- [46] Pierre Tandeo, Pierre Ailliot, Marc Bocquet, Alberto Carrassi, Takemasa Miyoshi, Manuel Pulido, and Yicun Zhen. A review of innovation-based methods to jointly estimate model and observation error covariance matrices in ensemble data assimilation. *Monthly Weather Review*, 148(10) :3973–3994, 2020.
- [47] Pierre Tandeo, Manuel Pulido, and François Lott. Offline parameter estimation using enkf and maximum likelihood error covariance estimates : Application to a subgrid-scale orography parametrization. *Quarterly journal of the royal meteorological society*, 141(687) :383–395, 2015.
- [48] Genta Ueno, Tomoyuki Higuchi, Takashi Kagimoto, and Naoki Hirose. Maximum likelihood estimation of error covariances in ensemble-based filters and its application to a coupled atmosphere–ocean model. *Quarterly Journal of the Royal Meteorological Society*, 136(650) :1316–1343, 2010.
- [49] Peter Jan Van Leeuwen. Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12) :4089–4114, 2009.

- [50] Peter Jan Van Leeuwen, Hans R Künsch, Lars Nerger, Roland Potthast, and Sebastian Reich. Particle filters for high-dimensional geoscience applications : A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723) :2335–2365, 2019.
- [51] Jeffrey S Whitaker and Thomas M Hamill. Ensemble data assimilation without perturbed observations. *Monthly weather review*, 130(7) :1913–1924, 2002.
- [52] Yue Ying, Fuqing Zhang, and Jeffrey L Anderson. On the selection of localization radius in ensemble filtering for multiscale quasigeostrophic dynamics. *Monthly Weather Review*, 146(2) :543–560, 2018.
- [53] Sheng Zhang and Guang Lin. Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A : Mathematical, Physical and Engineering Sciences*, 474(2217) :20180305, 2018.
- [54] Dusanka Zupanski and Milija Zupanski. Model error estimation employing an ensemble data assimilation approach. *Monthly Weather Review*, 134(5) :1337–1354, 2006.