Localisation des méthodes d'assimilation de donnée d'ensemble

Alban Farchi

On the localisation of ensemble data assimilation methods

Alban Farchi

On the localisation of ensemble data assimilation methods

École doctorale Sciences, Ingénierie et Environnement

On the localisation of ensemble data assimilation methods

Alban Farchi

Thèse de doctorat, spécialité physique

présentée le 21 novembre 2019

devant le jury composé de

Pr Alberto Carrassi	University of Utrecht	*Rapporteur*
Dr Emmanuel Cosme	Université Grenoble Alpes	*Rapporteur*
Dr Étienne Mémin	INRIA	*Président*
Dr Olivier Talagrand	CNRS	*Examinateur*
Pr Valérie Monbet	Université Rennes 1	*Examinatrice*
Dr Massimo Bonavita	ECMWF	*Examinateur*
Pr Marc Bocquet	École des Ponts ParisTech	*Directeur de thèse*

Remerciements

Cette thèse représente l'aboutissement de trois années de travail, un travail personnel, mais qui n'aurait pas été aussi abouti sans les contributions de nombreuses personnes. Pour commencer, je remercie sincèrement Alberto Carrassi et Emmanuel Cosme pour avoir accepté la charge de rapporteur et pour leur relecture attentive de mon manuscrit. Leurs questions et remarques pertinentes sur mon travail ont grandement contribué à son amélioration. Plus généralement, je remercie Étienne Mémin, Olivier Talagrand, Valérie Monbet et Massimo Bonavita pour leur participation à mon jury de thèse et pour les échanges constructifs que nous avons eus.

Je remercie chaleureusement Marc Bocquet, mon directeur de thèse. Il a su se rendre disponible pour répondre à mes questions, me donner des conseils quand j'en ai eu besoin et surtout pour m'aider à orienter mes recherches. Ce fût un plaisir de travailler avec lui car nous nous sommes très bien entendus et j'espère que cela sera toujours le cas lors de nos futures collaborations.

Faire de la recherche au sein du corps des Ponts n'est pas une chose facile. Un grand merci à toutes les personnes qui ont rendu cela possible en m'accompagnant lors de la préparation de mon dossier de thèse.

L'environnement de travail est un élément important dans la réussite. Aussi, je souhaiterais remercier l'ensemble des membres du CEREA, une équipe fort sympathique avec qui il est très agréable de travailler. En particulier, Yelva qui a su me guider dans les méandres de Polyphemus, Youngseob pour son soutien technique, Lydie qui a toujours été disponible pour m'aider, Joffrey pour les discussions passionnantes au bureau, mais aussi pour les partie de jeu en dehors, et tous les autres pour les bons moments passés ensemble. Rassurez-vous, ce n'est pas terminé puisque je rejoins le laboratoire de façon permanente !

Pour finir, je souhaite remercier de tout cœur mes proches, famille et amis, à qui je dédie cette thèse. Je suis reconnaissant à mes parents pour m'avoir soutenu et fait confiance toutes ces années. Mention particulière à ma maman pour sa très bonne organisation du pot de thèse et de la réunion de famille qui a suivi. Cela m'a bien soulagé de ne pas avoir à m'en occuper ! Merci à tout ceux qui ont fait le déplacement pour venir m'écouter et m'encourager lors de ma soutenance, même lorsque le sujet ne leur parlait pas beaucoup ! Enfin, merci à Virginie qui partage ma vie malgré l'éloignement. Merci pour m'avoir supporté lors de la rédaction du manuscrit, pour être présente quand j'en ai besoin, et surtout pour tous les moments heureux que nous passons ensemble.[1]

[1] Et merci pour avoir accepté de relire et corriger ce chapitre !

Contents

Résumé

Introduction – l'assimilation de données

En météorologie et en océanographie, un des objectifs est de prévoir l'état de l'écoulement atmosphérique et océanique. L'écoulement suit un certain nombre de lois physiques, par exemple les lois de conservation pour l'énergie et la masse. On désigne par modèle numérique l'ensemble des méthodes numériques mises en œuvre pour intégrer en temps une version discrétisée de ces lois. En météorologie et en océanographie, les modèles sont caractérisés par un très grand nombre de variables, ainsi que par de nombreuses incertitudes. La qualité des prédictions dépend principalement de trois éléments : l'adéquation entre les lois physiques et le modèle numérique, la qualité du modèle numérique (résolution et schémas d'intégration en particulier) et la précision des conditions initiales et des conditions de bord.

L'assimilation des observations est le procédé qui consiste à utiliser toutes les informations disponibles afin d'améliorer la précision des conditions initiales. L'assimilation de données est définie comme la discipline regroupant l'ensemble des méthodes mathématiques pouvant être utilisées pour assimiler les observations. Les techniques d'assimilation sont implémentées dans les centres opérationnels depuis plusieurs décennies et elles ont largement contribué à améliorer la qualité des prédictions.

L'assimilation de données d'ensemble et la localisation

En géosciences, la dimension des problèmes à résoudre est en général très grande. Il est donc nécessaire de développer des méthodes réduites. En s'inspirant des méthodes de Monte Carlo développées en statistiques, des méthodes d'ensemble ont été proposées pour l'assimilation de données. Dans ce contexte, la connaissance de l'état du système est décrite par un ensemble (en général quelques dizaines) de trajectoires du modèle. Les incertitudes sont alors naturellement décrites au moyen des propriétés statistiques de l'ensemble. Parmi les méthodes d'assimilation de données d'ensemble, deux classes se distinguent par leur popularité : le filtre de Kalman d'ensemble (EnKF) et le filtre particulaire (PF). L'EnKF est construit comme une variante ensembliste du fameux filtre de Kalman. En particulier, il s'appuie sur les mêmes hypothèses : linéarité des modèles et variables aléatoires Gaussiennes. Ce n'est pas le cas du PF qui s'appuie uniquement sur les méthodes de Monte Carlo et qui donc ne nécessite pas d'hypothèse supplémentaire.

En l'état, aucune de ces classes de méthodes ne fonctionne pour des systèmes de grande dimension. En effet, il semble impossible de représenter des propriétés statistiques complexes entre plusieurs milliards de variables en utilisant seulement quelques dizaines de membres. Cependant, dans la plupart des systèmes géophysiques, les corrélations décroissent très rapidement en fonction de la distance physique. Cette propriété est exploitée dans l'EnKF pour rendre le processus d'assimilation local ou bien pour corriger de manière artificielle les matrices de covariances empiriques.

La première méthode, qu'on appelle localisation par domaines, consiste à réaliser une collection d'analyses locales et indépendantes. L'analyse globale s'obtient en recollant les analyses locales. Les algorithmes ainsi construits, comme par exemple le *local ensemble transform Kalman filter*, sont très efficaces. Cependant, cette méthode ne permet pas d'assimiler des observations non-locales, comme les observations satellites, sans recourir à des approximations drastiques.

La deuxième méthode, qu'on appelle localisation des covariances, consiste à réaliser une analyse unique avec une matrice de covariance empirique localisée. En pratique, cette méthode est plus difficile à mettre en place dans un contexte déterministe, mais elle peut être utilisée telle quelle pour assimiler des observations non-locales. Ces dernières années, le nombre d'observations satellites devient de plus en plus important, il est donc nécessaire de développer des méthodes efficaces pour appliquer la localisation des covariances. Ce point est étudié dans la troisième partie de cette thèse.

En géophysique, les dynamiques sont en général non-linéaires et les distributions non-Gaussiennes. Il est donc souhaitable de développer l'assimilation de données au delà du cadre linéaire et Gaussien de l'EnKF, par exemple en utilisant le PF. D'un point de vue théorique, la localisation par domaines peut être directement appliquée au PF. En pratique, l'implémentation d'un PF localisé est un défi, puisque dans ce contexte il n'y a pas de méthode triviale pour recoller les analyses locales. La mise en œuvre de la localisation dans le PF est l'objet de la deuxième partie de cette thèse.

La localisation dans le filtre particulaire

Le principal atout du PF est qu'il permet de sortir du cadre linéaire et Gaussien. Le PF repose sur l'application successive de plusieurs étapes d'échantillonnage d'importance. En général, le coût de calcul de cette méthode croît exponentiellement avec la taille du système. C'est ce qu'on appelle la malédiction de la dimensionalité. En l'état, le PF ne peut donc pas être appliqué à des problèmes de grande dimension. Les techniques de localisation permettent de contourner la malédiction de la dimensionalité. Toutefois, l'implémentation de la localisation dans le PF soulève deux questions majeures : comment recoller des particules qui ont été mises à jour de façon locale, et comment limiter le déséquilibre dans l'ensemble recollé.

Nous proposons une classification théorique des filtres particulaires locaux (LPF) en deux catégories. Pour chaque catégorie, nous présentons les défis soulevés par l'implémentation de la localisation dans le PF et nous présentons l'ensemble des idées qui permettent la mise en œuvre pratique des algorithmes. Certaines de ces idées, d'ores et déjà dans la littérature, sont détaillées et parfois généralisées, tandis que d'autres sont nouvelles dans le domaine et contribuent à l'amélioration de la conception des algorithmes.

Dans la première classe d'algorithmes, on introduit la localisation dans l'analyse en faisant varier les poids des particules en fonction du point de grille. On obtient ensuite l'ensemble d'analyse globale en assemblant les ensembles mis à jour localement. La qualité de l'analyse dépend directement de la régularité de la méthode de mise à jour. Un ensemble de mauvaise qualité se traduit par des discontinuités, et donc du *déséquilibre*, dans l'ensemble d'analyse.

Nous présentons différentes méthodes qui permettent d'améliorer l'analyse en réduisant les discontinuités non-physiques. Parmi ces méthodes, les plus prometteuses reposent sur la théorie du transport optimal. Dans la seconde classe d'algorithmes, les observations sont assimilées une par une. On introduit la localisation de façon plus générale, au moyen d'une partition des variables. L'objectif de cette partition est de construire un cadre d'application pour la localisation en s'affranchissant du problème des discontinuités. Nous présentons ensuite deux méthodes qui permettent d'implémenter ce type de localisation.

Nous avons implémenté et testé de manière systématique les algorithmes LPF en utilisant des expériences jumelles avec des modèles de petite taille : le modèle de Lorenz 1996 avec 40 variables et un modèle bidimensionnel basé sur l'équation de la vorticité barotropique avec 1024 variables. Dans ces deux modèles, les algorithmes LPF sont simples à implémenter et fonctionnent comme prévu. Les scores obtenus sont acceptables, même en utilisant des ensembles de petite taille, alors que l'on se situe dans des régimes de fonctionnement où le PF global est dégénéré. Dans tous les cas testés, nous avons constaté que les algorithmes qui utilisent le transport optimal obtiennent des scores significativement meilleurs que ceux des autres algorithmes. Nous interprétons ce résultat comme une démonstration de la réduction des discontinuités non-physiques dans l'ensemble d'analyse. De plus, dans une configuration faiblement non-linéaire pour le modèle de Lorenz 1996, les meilleurs scores obtenus par les algorithmes LPF sont meilleurs que ceux de l'EnKF de référence. Nous avons ensuite implémenté les algorithmes LPF dans une configuration à haute résolution du modèle de vorticité barotropique avec 65 536 variables. Les résultats obtenus avec ce modèle confirment ceux obtenus avec les modèles de petite taille et montrent que les algorithmes LPF sont suffisamment matures pour être appliqués à des systèmes géophysiques de grand dimension.

Enfin, nous nous penchons sur le cas de la prévision des concentrations d'ozone dans la troposphère en Europe de l'ouest pendant l'été 2009. Nous avons à notre disposition des mesures horaires de concentration d'ozone en plusieurs centaines de stations. Pour ce jeu d'expériences, nous utilisons le modèle `Polair3DChemistry` de la plateforme `Polyphemus`. Le modèle est débiaisé en utilisant une paramétrisation simple. La simulation de référence débiaisée, comparée aux observations, permet d'obtenir des scores du même ordre de grandeur que la plupart des modèles en chimie atmosphérique. Nous expliquons ensuite comment mettre en œuvre l'assimilation de données dans ce système, au moyen de différents algorithmes dont plusieurs algorithmes LPF. Les résultats obtenus soulignent le bon fonctionnement des algorithmes : les scores de vérification sont significativement meilleurs que ceux de la simulation de référence. Nous montrons que les algorithmes LPF obtiennent des scores très similaires à ceux de l'EnKF de référence, ce qui est une première pour un système géophysique de grande dimension. Dans nos expériences, les algorithmes d'assimilation d'ensemble semblent avoir le dessus sur l'interpolation optimale (algorithme sans ensemble). Cependant, il n'est pas évident que le faible gain dans les scores de validation soit suffisant pour justifier l'énorme augmentation du temps de calcul liée à la prévision d'ensemble.

La localisation des covariances dans le filtre de Kalman d'ensemble

La localisation des covariances est la seule méthode de localisation qui permet d'assimiler des observations non-locales de façon rigoureuse. Dans cette partie, nous commençons par explorer

différentes techniques qui permettent de mettre en œuvre la localisation des covariances au moyen d'un ensemble augmenté. Nous discutons des deux principales difficultés liées à cette approche : comment construire l'ensemble augmenté et comment mettre à jour l'ensemble.

Nous présentons deux méthodes différentes pour construire l'ensemble augmenté. La première méthode repose sur une propriété de factorisation et est déjà répandue en assimilation de données pour les géosciences. La deuxième méthode est une approche alternative que nous proposons. Cette approche repose sur l'utilisation de techniques *randomisées* pour le calcul des décompositions en valeurs singulières. Ces techniques sont très efficaces lorsque la matrice de localisation est simple à appliquer. Dans les deux cas, la mise à jour de l'ensemble se fait au moyen d'une formule simple d'algèbre linéaire dans l'espace de l'ensemble augmenté. Les méthodes sont testées et comparées en utilisant des expériences jumelles avec le modèle de Lorenz 1996 avec 400 variables. Dans ce problème, nous montrons que la seconde méthode, celle qui utilise les techniques *randomisées*, permet d'obtenir de meilleurs scores que la première méthode en utilisant un ensemble augmenté de plus petite taille.

L'EnKF avec ensemble augmenté est ensuite généralisé au cas de l'assimilation d'observations satellites dans un modèle étendu en espace. Dans ce cas, la localisation des covariances est utilisée dans la direction verticale et la localisation par domaines est utilisée dans la direction horizontale. L'algorithme généralisé est mis en œuvre et testé au moyen d'expériences jumelles avec une extension à plusieurs couches du modèle de Lorenz 1996. Cette extension possède un total de 1280 variables qui sont observées en utilisant un opérateur d'observation conçu pour imiter des radiances satellitaires. Comme on pouvait s'y attendre dans ce système avec des observations non-locales, notre algorithme généralisé obtient de bien meilleurs scores que l'EnKF de référence, pour lequel seule la localisation par domaines est mise en place.

Dans un deuxième temps, nous étudions la cohérence de la mise à jour des perturbations dans l'EnKF déterministe qui utilise la localisation des covariances. Nous montrons que dans ce cas, les perturbations d'analyse ne représentent pas les modes principaux de la matrice de covariance d'analyse, contrairement à ce qui se passe quand on utilise la localisation par domaines. Fort de ces considérations, nous proposons une nouvelle méthode de mise à jour des perturbation. Cette méthode, potentiellement plus cohérente, nécessite la résolution d'un problème d'optimisation. On s'attend alors à ce que les perturbations d'analyses permettent de mieux représenter les corrélations à courte portée dans la mesure où on exerce moins de contraintes sur les corrélations à longue portée. Il se trouve que le gradient de la fonction de coût se calcule au moyen d'une formule analytique, on peut donc utiliser un algorithme de minimisation itératif. Cependant, le calcul de la fonction de coût et de son gradient nécessite une connaissance partielle de la matrice de covariance d'analyse, ce qui peut représenter une difficulté dans la mise en œuvre de la méthode pour des systèmes de grande dimension.

La nouvelle méthode est testée et comparée à différents algorithmes EnKF de référence en utilisant des expériences jumelles de deux modèles de petite dimension : le modèle de Lorenz 1996 avec 40 variables et une version discrétisée dans l'espace spectral du modèle de Kuramoto–Sivashinsky avec 128 variables. Pour les deux modèles, nous montrons que le besoin d'inflation multiplicative est fortement réduit avec le nouvel algorithme. De plus, lorsque la taille de l'ensemble est suffisante, le nouvel algorithme obtient de très bons résultats sans inflation. Cela montre que la nouvelle méthode permet effectivement d'obtenir une meilleure cohérence entre la matrice de covariance empirique de l'ensemble d'analyse et la véritable matrice de covariance d'analyse. Ce résultat pourrait s'interpréter physiquement

comme la réduction du déséquilibre engendré par la localisation. De plus, nous montrons que l'utilisation de la nouvelle méthode permet d'obtenir des scores significativement meilleurs. Ces résultats ont été confirmés et renforcés par des expériences dans des configurations pour lesquelles le réseau d'observation est creux ou pour lesquelles la fréquence d'observation est plus faible.

Références

Bocquet, Marc et Alban Farchi (2019). ≪ On the consistency of the local ensemble square root Kalman filter perturbation update ≫. In : *Tellus A : Dynamic Meteorology and Oceanography* 71.1, p. 1–21.

Farchi, Alban et Marc Bocquet (2018). ≪ Review article : Comparison of local particle filters and new implementations ≫. In : *Nonlinear Processes in Geophysics* 25.4, p. 765–807.

— (2019). ≪ On the efficiency of covariance localisation of the ensemble Kalman filter using augmented ensembles ≫. In : *Frontiers in Applied Mathematics and Statistics* 5.

Conventions

Sets	
$\mathbb{R}$	real numbers
$\mathbb{R}_+$	nonnegative real numbers
$\mathbb{R}_+^*$	positive real numbers
$\mathbb{N}$	positive integer numbers
$\mathbb{R}^n$	vectors with n elements
$\mathbb{R}^{n\times p}$	matrices with n rows and p columns
$\mathbb{F}^{n\leftarrow p}$	functions $\mathbb{R}^n \to \mathbb{R}^p$
$\mathbb{G}^{n\leftarrow p}$	functions $\mathbb{F}^{n\leftarrow p} \to \mathbb{F}^{n\leftarrow p}$
$\mathbb{G}^{n\leftarrow(p\leftarrow q)}$	functions $\mathbb{F}^{p\leftarrow q} \to \mathbb{R}^n$
$\operatorname{Card}\mathbb{A}$	cardinal of the finite set $\mathbb{A}$

Stylistic conventions

Object	Style	Example
Scalar	italic	x
Vector	lower-case bold roman	$\mathbf{x}$
Matrix	upper-case bold roman	$\mathcal{M}$
Function	upper-case calligraphic	$\mathcal{F}$
Ensemble	upper-case sans-serif	E
Random vector	lower-case bold italic	$\boldsymbol{x}$

Sequences	
$(j:i)$	sequence of integers $(i,\ldots,j)$, with $i \leq j$
$(j:)$	sequence of integers $(0,\ldots,j)$
$X_{j:i}$	sequence of objects $(X_i,\ldots,X_j)$ [*scalars, vectors, or matrices*]
$X_{j:}$	sequence of objects $(X_0,\ldots,X_j)$ [*scalars, vectors, or matrices*]
$\mathcal{O}$	indicates a domination relationship
$\sim$	indicates an equivalence relationship

Linear algebra	
$[\mathbf{x}]_n$	n-th element of the vector $\mathbf{x}$
$[\mathbf{M}]_{n,p}$	n-th row, p-th column element of the matrix $\mathbf{M}$
$\mathbf{0}$	vector filled with zeros
$\mathbf{1}$	vector filled with ones
$\mathbf{I}$	identity matrix
$\det \mathbf{M}$	determinant of the matrix $\mathbf{M}$
$\operatorname{tr} \mathbf{M}$	trace of the matrix $\mathbf{M}$
$\operatorname{rk} \mathbf{M}$	rank of the matrix $\mathbf{M}$
$\circ$	element-wise multiplication for vectors or matrices [*to be defined*]
$\lVert\mathbf{x}\rVert_2$	$\mathcal{L}^2$-norm of the vector $\mathbf{x}$
$\lVert\mathbf{M}\rVert_{\mathrm{F}}$	Frobenius norm of the matrix $\mathbf{M}$
$\operatorname{diag}(\mathbf{M})$	vector containing only the diagonal elements of the matrix $\mathbf{M}$
$\operatorname{vec}(\mathbf{M})$	vector containing all the elements of the matrix $\mathbf{M}$
Δ	modulation product for matrices [*to be defined*]
$\otimes$	Kronecker product
$\sigma_n(\mathbf{M})$	n-th singular value of the matrix $\mathbf{M}$

Functions and differential calculus	
$\circ$	composition operator for functions
$\nabla\mathcal{F}\vert_{\mathbf{x}}$	gradient of the function $\mathcal{F}$, evaluated at $\mathbf{x}$
$\operatorname{Hess}\mathcal{F}\vert_{\mathbf{x}}$	Hessian matrix of the function $\mathcal{F}$, evaluated at $\mathbf{x}$
div	divergence operator
Δ	Laplacian operator

Standard sizes and associated iterators

Symbol	Description	Iterators
N_{x}	dimension of the state space	n, m
N_{y}	dimension of the observation space	p, q
N_{e}	ensemble size	i, j
N_{t}	number of threads for parallel computing	
N_{eff}	effective ensemble size [*to be defined*]	
$\widehat{N}_{\mathrm{e}}$	augmented ensemble size [*to be defined*]	
N_{m}	number of modes [*to be defined*]	

Time evolution	
t	time
k, l	time indices [subscripts], *omitted in the text unless necessary*
t_k	k-th observation time
N_c	total number of cycles in a simulation
N_s	number of spin-up cycles in a simulation

Probabilities	
$\nu[\boldsymbol{x}]$	distribution of the random vector $\boldsymbol{x}$
$\mathcal{N}[\mathbf{x}, \mathbf{P}]$	Gaussian (normal) distribution with mean $\mathbf{x}$ and covariance matrix $\mathbf{P}$
$\mathcal{U}[x, y]$	uniform distribution over the interval $[x, y]$
$\mathcal{LN}[\mathbf{x}, \mathbf{P}]$	Log-normal distribution with mean $\mathbf{x}$ and covariance matrix $\mathbf{P}$
$\pi[\boldsymbol{x}](\mathbf{x})$	pdf of the random vector $\boldsymbol{x}$, evaluated at $\mathbf{x}$
$\mathcal{N}(\mathbf{x}\vert\mathbf{y}, \mathbf{P})$	pdf of the Gaussian distribution $\mathcal{N}[\mathbf{y}, \mathbf{P}]$, evaluated at $\mathbf{x}$
$\boldsymbol{x} \sim \nu$	indicates that the random vector $\boldsymbol{x}$ has distribution ν
$x \sim \nu$	indicates that the scalar x is a random draw from the distribution ν
$\mathbf{x} \sim \nu$	indicates that the vector $\mathbf{x}$ is a random draw from the distribution ν
$\mathsf{E} \overset{\text{iid}}{\sim} \nu$	indicates that the ensemble E is an iid sample from ν
$\mathbb{E}[\boldsymbol{x}]$	expectation of the random vector $\boldsymbol{x}$
$\mathbb{V}[\boldsymbol{x}]$	variance of the random vector $\boldsymbol{x}$
δ	Dirac kernel

Decorations	
$\cdot \triangleq \cdot$	indicates a definition
$.^\mathsf{T}$	transposition operator
$.^+$	indicates a matrix pseudo-inverse
$.^{1/2}$	indicates a matrix square root [*to be defined*]
$\bar{\cdot}$	indicates a sample estimate [*to be defined*]
$.^*$	refers to some notion of optimality [*to be defined*]
$.^\ell$	refers to some notion of locality [*to be defined*]

Introduction

Meteorology and oceanography are the scientific disciplines which consist in the study of atmospheric and oceanic phenomena, with the aim of predicting the state of the atmospheric and oceanic flows. The prediction starts from the physical laws governing the flow, mainly the conservation laws for mass, energy and momentum. A model is then defined as the set of numerical methods used to integrate in time a discrete version of these laws. In meteorology and oceanography, the models are characterised by a very high number of state variables, and by many sources of uncertainty. The quality of the predictions mainly depends on three elements: the match between the physical laws described by the model and the actual physical laws, the numerical quality of the model (resolution, and integration methods), and the accuracy of the initial and boundary conditions.

The assimilation of observations can be defined as the method used to exploit all available information with the aim of providing to the model an initial condition as accurate as possible. An observation is defined here as the result of the measurement process of a physical quantity. Measurement processes vary significantly and yield observations of different nature, which in turn may require different assimilation methods. Data assimilation can be defined as the discipline gathering all mathematical methods used to assimilate the observations. Since the early days of numerical weather prediction, in the middle of the twentieth century, continuous progress in the theory, in the algorithms, and in the available computational resources have significantly contributed to the increase in quality of the forecasts.

As the dimension of typical problems in geophysical data assimilation is often very large, it is necessary to develop reduced method. Taking inspiration from the Monte Carlo methods in the statistical literature, ensemble methods have been proposed for data assimilation. In this case, the current knowledge on the system is described by an ensemble (typically a few dozen) of trajectories of the model. The statistical properties of the ensemble are used to quantify the uncertainties. Currently, the two most widespread classes of ensemble data assimilation methods are the ensemble Kalman filter (EnKF) and the particle filter (PF). The EnKF has been designed as an ensemble variant of the famous Kalman filter and hence it relies on the same set of assumptions about linearity and Gaussianity. By contrast, the PF only relies on Monte Carlo methods, and therefore it does not require additional assumptions.

As is, both classes of methods cannot be applied to high-dimensional systems. Indeed, it seems impossible to represent complex statistical properties between several billions of state variables while using only a few dozen ensemble members. In most geophysical systems however, the correlation between spatially distant parts of the system decrease at a fast rate with the physical distance. This property is used in the EnKF to make the assimilation of observations local or, alternatively, to artificially taper the sample error covariance matrices. The first method is known as domain localisation, and the second as covariance localisation.

Domain localisation consists of a collection of local and independent ensemble updates. The whole updated ensemble is obtained by assembling the locally updated ensembles.

Furthermore, the transition between the updates of adjacent domains can be made smoother by tapering the precision of the attached observations. This leads to efficient data assimilation algorithms, for example the local ensemble transform Kalman filter, which has become an emblem of the success of the EnKF in high-dimensional geophysical systems. When using domain localisation however, satellite observations cannot be assimilated without *ad hoc* approximations. By contrast, covariance localisation consists of a single ensemble update using a localised forecast sample covariance matrix. This is in practice much less simple to implement in a deterministic context, but it can be used to assimilate satellite observations without further approximations. The huge increase of satellite observations in the recent years justify the need for efficient implementations of covariance localisation in the EnKF.

In geophysics, the dynamics is often nonlinear and the error distributions are in general non-Gaussian. Therefore, it would be desirable to develop data assimilation algorithms beyond the Gaussian and linear framework of the EnKF, for example with the PF. From a theoretical point of view, domain localisation could be used in the PF to make it applicable to high-dimensional systems. However, the implementation of localisation in the PF is a challenge, because in this context there is no trivial way of gluing locally updated ensembles together. Recent developments in particle filtering have been proposed to overcome this difficulty. The resulting local PF algorithms, very different from one another, have the potential to be applied to high-dimensional geophysical systems.

In this thesis, we study an improve localisation methods for ensemble data assimilation algorithms. The first part provides an overview of the filtering methods in data assimilation. Chapter 1 introduces the mathematical formalism. Chapter 2 describes the EnKF, and chapter 3 describes the PF. The second part is dedicated to the implementation of localisation in the PF. Chapter 4 is a review of the recent development in local particle filtering. A generic and theoretical classification of local PF algorithms is introduced, with an emphasis on the advantages and drawbacks of each category. Alongside the classification, practical solutions to the difficulties of local particle filtering are suggested. They lead to new implementations and improvements in the design of the local PF algorithms. Chapter 5 systematically tests and compares the local PF algorithms using twin experiments of low- to medium-order systems. Chapter 6 considers the case study of the prediction of the tropospheric ozone using concentration measurements. Several data assimilation algorithms, including local PF algorithms, are implemented and applied to this problem. Finally the third part is dedicated to the implementation of covariance localisation in the EnKF. Chapter 7 shows how covariance can be efficiently implemented in the deterministic EnKF using an augmented ensemble. The proposed algorithm is tested in particular using twin experiments of a medium-order model with satellite-like observations. Chapter 8 studies the consistency of the deterministic EnKF with covariance localisation. A new implementation is proposed, and compared to the original one using twin experiments of low-order models.

Part I

Filtering methods in ensemble data assimilation

1 Introduction to the methods of data assimilation

Contents

Data assimilation (DA) is the discipline which gathers all methods designed to improve the knowledge of the state (past, present or future) of a dynamical system using both observations and modelling results of this system. The most common application of DA in the geoscience is numerical weather prediction (NWP). In this case, the state variables include, among others, the pressure and temperature fields. Their evolution is governed by the Navier–Stokes equation, the first law of thermodynamics, and the ideal gas law. The observations mainly come from weather stations and from satellite instruments.

Classical methods in DA can be divided into two categories: the statistical approach and the variational approach. Only recent methods combine both approaches into an hybrid framework. This chapter gives an introduction to the most common methods in DA, and is inspired from the following references: Cohn (1997), Bocquet (2014), Law et al. (2015), Asch et al. (2016) and Carrassi et al. (2018). In section 1.1, we introduce the mathematical formalism for DA. In section 1.2, we formulate the estimation problem. The statistical and variational approaches are then briefly introduced in sections 1.3 and 1.4. Finally section 1.5 presents basic properties of the most simple DA methods.

1.1 Mathematical formalism for data assimilation

We start this chapter with an introduction about the mathematical formalism necessary for DA problems. For simplicity, it is assumed in this thesis that all random variables and random vectors have a probability density function (pdf) with respect to the Lebesgue measure.

1.1.1 The hidden Markov model

A generic discrete-time DA system is an hidden Markov model (HMM) which describes a random process $(\boldsymbol{x}_k, \boldsymbol{y}_k)_{k\in\mathbb{N}}$, for a strictly growing sequence of time instants $(t_k)_{k\in\mathbb{N}}$. The hidden **state vector** $\boldsymbol{x}$, a random vector with N_{x} elements, is only known through the **observation vector** $\boldsymbol{y}$, a random vector with N_{y} elements. The Markov property of the model ensures that the pdf of the joint distribution can be factored as[1]

$$\pi[\boldsymbol{x}_{k:}, \boldsymbol{y}_{k:}] = \pi[\boldsymbol{x}_0]\pi[\boldsymbol{y}_0|\boldsymbol{x}_0]\prod_{l=1}^{k}\pi[\boldsymbol{x}_l|\boldsymbol{x}_{l-1}]\pi[\boldsymbol{y}_l|\boldsymbol{x}_l]. \tag{1.1}$$

This means that the model is entirely determined by the **background** density $\pi^{\mathsf{b}} \triangleq \pi[\boldsymbol{x}_0]$, the observation density $\pi^{\mathsf{o}}_k \triangleq \pi\big[\boldsymbol{y}_k\big|\boldsymbol{x}_k\big]$, and the transition density $\pi^{\mathsf{m}}_k \triangleq \pi\big[\boldsymbol{x}_{k+1}\big|\boldsymbol{x}_k\big]$. This is summarised by the following system.

Generic HMM

$$\boldsymbol{x}_0 \sim \nu^{\mathsf{b}}, \tag{1.2a}$$
$$\boldsymbol{y}_k = \nu^{\mathsf{o}}_k, \tag{1.2b}$$
$$\boldsymbol{x}_{k+1} = \nu^{\mathsf{m}}_k. \tag{1.2c}$$

In this system, $\nu^{\mathsf{b}} \triangleq \nu[\boldsymbol{x}_0]$ is the background distribution, and

$$\nu^{\mathsf{o}}_k \triangleq \nu\big[\boldsymbol{y}_k\big|\boldsymbol{x}_k\big], \tag{1.3}$$
$$\nu^{\mathsf{m}}_k \triangleq \nu\big[\boldsymbol{x}_{k+1}\big|\boldsymbol{x}_k\big], \tag{1.4}$$

are the observation and the transition distributions, most of the time defined using a dynamical **model** and an **observation operator**. These elements are discussed in subsections 1.1.2 and 1.1.3. Figure 1.1 illustrates the generic HMM, system 1.2.

The starting point in DA is a realisation of the HMM, written $(\mathbf{x}^{\mathsf{t}}_k, \mathbf{y}_k)_{k\in\mathbb{N}}$. The goal is to estimate, in a sense that is defined in section 1.2, the **truth** $\mathbf{x}^{\mathsf{t}}$, with the only data being

[1] Unless specified otherwise, all equations referring to the time index k are valid for all $k \in \mathbb{N}$.

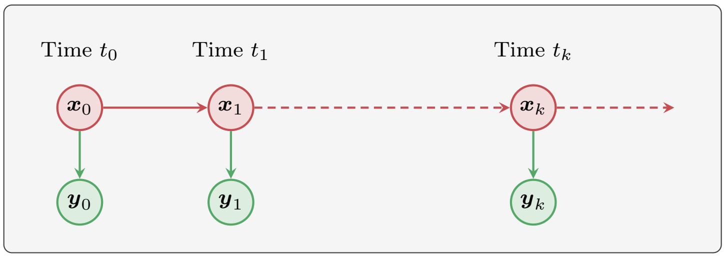

Figure 1.1: Schematic representation of the generic HMM, system 1.2. The hidden part is in red and the observed part in green.

the observation vector $\mathbf{y}$.[2] In that sense, DA belongs to the class of the so-called inverse problems. There are three cases:

1. estimating $\mathbf{x}^{\mathsf{t}}_{k+l}$ from $\mathbf{y}_{k:}$, with $l > 0$, is called **prediction**;
2. estimating $\mathbf{x}^{\mathsf{t}}_{k}$ from $\mathbf{y}_{k:}$ is called **filtering**;
3. estimating $\mathbf{x}^{\mathsf{t}}_{k-l}$ from $\mathbf{y}_{k:}$, with $l \geq 0$, is called **smoothing**.

In this thesis, the focus is on filtering. Prediction and smoothing problems are not discussed, although prediction skills can be evaluated.

1.1.2 The dynamical system

For a complex dynamical system, the state cannot be entirely described by a finite-dimensional vector. Let us consider the example of meteorology. The state of the system is a three-dimensional vector field $\mathbf{x}^{\dagger}$ which describes the spatial evolution evolution of the N relevant variables for meteorology: temperature, pressure... Assume that the temporal evolution of $\mathbf{x}^{\dagger}$ is given by

$$\mathbf{x}^{\dagger}(t_{k+1}) = \mathcal{M}^{\dagger}\big(\mathbf{x}^{\dagger}(t_k)\big), \tag{1.5}$$

where the model $\mathcal{M}^{\dagger}$ could be, for example, the resolvent of the equations of meteorology.[3]

From a numerical point of view, it is necessary to tabulate the values of $\mathbf{x}^{\dagger}$. That is, the vector space of three-dimensional fields with N variables $\mathbb{F}^{N\leftarrow 3}$, an infinite-dimensional vector space, is represented by $\mathbb{R}^{N_{\mathrm{x}}}$, an N_{x}-dimensional vector space, through a projection operator $\mathbf{\Pi}$. For example, $\mathbf{\Pi}$ could be a set of averaging operators. For this system, the truth $\mathbf{x}^{\mathsf{t}}$ is defined as

$$\mathbf{x}^{\mathsf{t}}_k \triangleq \mathbf{\Pi}\big(\mathbf{x}^{\dagger}(t_k)\big). \tag{1.6}$$

This is obviously an abuse of language, because the true state of the system is indeed $\mathbf{x}^{\dagger}$. However, since it is impossible to even represent $\mathbf{x}^{\dagger}$,[4] we focus on the problem which consists in estimating $\mathbf{x}^{\mathsf{t}}$.

[2] Sans-serif exponents are used to distinguish between different variants of the state vector $\mathbf{x}$. By contrast, the observation vector $\mathbf{y}$ is given and there is no need to make a distinction. The term *observation vector* indiscriminately refers to the vector $\mathbf{y}$ or to the random vector $\boldsymbol{y}$.

[3] The question of the existence and of the time dependence of such an evolution model for the equations of meteorology is set apart.

[4] Unless there are analytic formulae for $\mathbf{x}^{\dagger}$, which highly limits the possibilities.

In the modelling (state) space $\mathbb{R}^{N_\mathsf{x}}$, the model $\mathcal{M}$ is determined by the numerical scheme which is used to integrate the equations of meteorology in $\mathbb{R}^{N_\mathsf{x}}$. The model error $\mathbf{e}^\mathsf{m}$ is then defined as

$$\mathbf{e}^\mathsf{m}_{k+1} \triangleq \mathbf{x}^\mathsf{t}_{k+1} - \mathcal{M}\big(\mathbf{x}^\mathsf{t}_k\big). \tag{1.7}$$

From the relationship between $\mathbf{x}^\mathsf{t}$ and $\mathbf{x}^\dagger$, equation (1.6), we deduce that

$$\mathbf{e}^\mathsf{m}_{k+1} = \Big(\mathbf{\Pi} \circ \mathcal{M}^\dagger - \mathcal{M} \circ \mathbf{\Pi}\Big)\Big(\mathbf{x}^\dagger\big(t_k\big)\Big). \tag{1.8}$$

This means that the model error $\mathbf{e}^\mathsf{m}$ is composed of some representation error, inherent to the projection $\mathbf{\Pi}$, and of some potential modelling error, when the model $\mathcal{M}$ do not match the discretisation of the model $\mathcal{M}^\dagger$.

Obviously, the exact model error $\mathbf{e}^\mathsf{m}$ is unknown. However, it can be seen as the realisation of a random vector $\boldsymbol{e}^\mathsf{m}$. Back to the HMM, this means that the random vectors $\boldsymbol{x}_{k+1}$ and $\boldsymbol{x}_k$ are related by

$$\boldsymbol{x}_{k+1} = \mathcal{M}\big(\boldsymbol{x}_k\big) + \boldsymbol{e}^\mathsf{m}_k, \tag{1.9}$$

and that the transition density is characterised by the pdf of the random vector $\boldsymbol{e}^\mathsf{m}$.

Remark 1. In most DA applications, the pdf of the model error $\pi[\boldsymbol{e}^\mathsf{m}]$ is given. However, the problem which consists in estimating the distribution of the unknown model error $\mathbf{e}^\mathsf{m}$ for a complex dynamical system is usually non-trivial.

1.1.3 The observation system

For a geophysical system, there are two classes of observations. We speak of **conventional observation** when the measurement is performed on a station, either on the surface or on board a vehicle (aircraft, ship or sounding balloon). The spatial coverage of these observations is therefore sparse, but the measurement delivers a direct information on the system, for example, the temperature at a specific spatial location and at a specific time. In that sense, conventional observations are **local**.

We speak of **remote sensing** or **satellite observation** when the measurement is performed by a distant instrument, often a satellite (radar, lidar, . . .). Contrary to the conventional observations, the spatial coverage of satellite observations is dense and redundant, however the measurement delivers an indirect information on the system, for example, the radiance of a whole column of air. In that sense, satellite observations are **non-local** and hence harder to assimilate. In the last decades, the number of satellite observations has blown-up, in such a way that nowadays, the overwhelming majority of observations comes from satellite measurements.

From a mathematical point of view, let $\mathcal{H}^\dagger$ be the map that describes the processes involved in the measurement: evaluation, average. . . With a perfect instrument, the observation vector would be given by

$$\mathbf{y}_k = \mathcal{H}^\dagger\big(\mathbf{x}^\dagger(t_k)\big). \tag{1.10}$$

The **instrumental error $\mathbf{e}^\mathsf{i}$** is defined as

$$\mathbf{e}^\mathsf{i}_k \triangleq \mathbf{y}_k - \mathcal{H}^\dagger\big(\mathbf{x}^\dagger(t_k)\big). \tag{1.11}$$

In the modelling space $\mathbb{R}^{N_\mathsf{x}}$, the observation operator $\mathcal{H}$ is determined by the numerical scheme which is used to represent the measurement of a state vector. The observation error $\mathbf{e}^{\mathsf{o}}$ is then defined as

$$\mathbf{e}^{\mathsf{o}}_k \triangleq \mathbf{y}_k - \mathcal{H}(\mathbf{x}^{\mathsf{t}}_k). \tag{1.12}$$

From the relationship between $\mathbf{y}$ and $\mathbf{x}^\dagger$, equation (1.11), and the relationship between $\mathbf{x}^{\mathsf{t}}$ and $\mathbf{x}^\dagger$, equation (1.6), we deduce that

$$\mathbf{e}^{\mathsf{o}}_k = \Big(\mathcal{H}^\dagger - \mathcal{H} \circ \mathbf{\Pi}\Big)\Big(\mathbf{x}^\dagger(t_k)\Big) + \mathbf{e}^{\mathsf{i}}_k. \tag{1.13}$$

This means that the observation error $\mathbf{e}^{\mathsf{o}}$ is composed of some representation error, inherent to the projection $\mathbf{\Pi}$, of some potential modelling error, when the observation operator $\mathcal{H}$ do not match the discretisation of the map $\mathcal{H}^\dagger$, and of the instrumental error $\mathbf{e}^{\mathsf{i}}$.

Again, the exact observation error $\mathbf{e}^{\mathsf{o}}$ is unknown and can be seen as the realisation of a random vector $\boldsymbol{e}^{\mathsf{o}}$. Back to the HMM, this means that the random vectors $\boldsymbol{y}_k$ and $\boldsymbol{x}_k$ are related by

$$\boldsymbol{y}_k = \mathcal{H}\big(\boldsymbol{x}_k\big) + \boldsymbol{e}^{\mathsf{o}}_k, \tag{1.14}$$

and that the observation density is characterised by the pdf of the random vector $\boldsymbol{e}^{\mathsf{o}}$.

Remark 2. In most DA applications, the pdf of the observation error $\pi[\boldsymbol{e}^{\mathsf{o}}]$ is given. However, the problem which consists in estimating the distribution of the unknown observation error $\mathbf{e}^{\mathsf{o}}$ for complex dynamical- and observation systems is usually non-trivial.

1.1.4 The generic DA system

By bringing together all the pieces, we obtain the following formulation for the generic DA system.

Generic DA system

$$\boldsymbol{x}_0 \sim \nu^{\mathsf{b}}, \tag{1.15a}$$

$$\boldsymbol{y}_k = \mathcal{H}\big(\boldsymbol{x}_k\big) + \boldsymbol{e}^{\mathsf{o}}_k, \qquad \boldsymbol{e}^{\mathsf{o}}_k \sim \nu\big[\boldsymbol{e}^{\mathsf{o}}_k\big], \tag{1.15b}$$

$$\boldsymbol{x}_{k+1} = \mathcal{M}\big(\boldsymbol{x}_k\big) + \boldsymbol{e}^{\mathsf{m}}_k, \qquad \boldsymbol{e}^{\mathsf{m}}_k \sim \nu\big[\boldsymbol{e}^{\mathsf{m}}_k\big]. \tag{1.15c}$$

Equation (1.15b) may be called the observation equation and equation (1.15c) the transition equation. This generic DA system is illustrated in figure 1.2.

Remark 3. In the generic DA system, the dynamical model $\mathcal{M}$ and the observation operator $\mathcal{H}$ do not depend on the time index k. The generalisation to time-dependent $\mathcal{M}_{k+1:k}$ and $\mathcal{H}_k$ is straightforward in this thesis.

1.2 The filtering estimation problem

In this section, we formulate the filtering estimation problem.

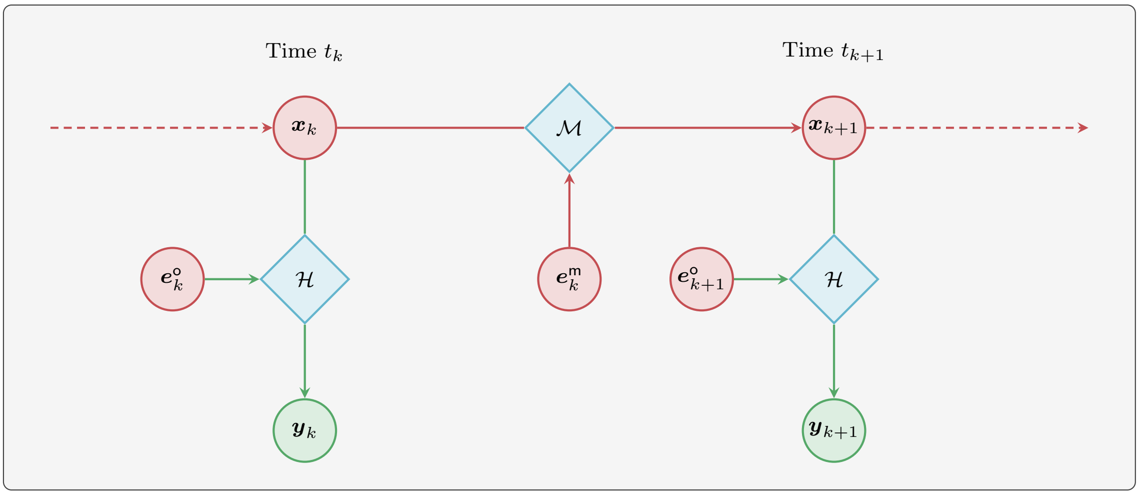

Figure 1.2: Schematic representation the generic DA system, system (1.15). The hidden part is in red, the observed part in green, and the maps are in cyan.

1.2.1 The generic filtering estimation problem

Let π^{a} be the filtering density defined as

$$\pi^{\mathsf{a}}_k \triangleq \pi\big[\boldsymbol{x}_k \big| \boldsymbol{y}_{k:}\big]. \tag{1.16}$$

The filtering density π^{a} gathers all available information at a given time about the unknown truth $\mathbf{x}^{\mathsf{t}}$. This is why the ultimate goal in filtering DA should be to compute π^{a}. This is formalised in problem 1.1.

Problem 1.1 (Generic filtering estimation problem)**.** *Given the sequence of observation vectors* $\mathbf{y}$*, compute the filtering density* π^{a} *of the generic DA system.*

Let π^{f} be the prediction density, defined as

$$\pi^{\mathsf{f}}_{k+1} \triangleq \pi\big[\boldsymbol{x}_{k+1} \big| \boldsymbol{y}_{k:}\big]. \tag{1.17}$$

The prediction operator $\mathcal{P}$ is then defined as

$$\mathcal{P}_k(\pi)(\mathbf{x}_{k+1}) \triangleq \int \pi^{\mathsf{m}}_k(\mathbf{x}_{k+1}|\mathbf{x}_k)\,\pi(\mathbf{x}_k)\,\mathrm{d}\mathbf{x}_k, \tag{1.18}$$

where the integral is taken over the whole state space $\mathbb{R}^{N_{\mathrm{x}}}$. And finally the correction operator $\mathcal{C}$ is defined as

$$\mathcal{C}_0(\pi)(\mathbf{x}_0) \triangleq \frac{\pi^{\mathsf{o}}_0(\mathbf{y}_0|\mathbf{x}_0)\,\pi(\mathbf{x}_0)}{\pi[\boldsymbol{y}_0](\mathbf{y}_0)}, \tag{1.19a}$$

$$\mathcal{C}_{k+1}(\pi)(\mathbf{x}_{k+1}) \triangleq \frac{\pi^{\mathsf{o}}_{k+1}(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\,\pi(\mathbf{x}_{k+1})}{\pi[\boldsymbol{y}_{k+1}|\boldsymbol{y}_{k:}](\mathbf{y}_{k+1}|\mathbf{y}_{k:})}. \tag{1.19b}$$

With these definitions and using the Markov property of the system, the Chapman–Kolmogorov equation is

$$\pi^{\mathsf{f}}_{k+1} = \mathcal{P}_k\big(\pi^{\mathsf{a}}_k\big), \tag{1.20}$$

and Bayes' theorem is written

$$\pi^{\mathsf{a}}_0 = \mathcal{C}_0\big(\pi^{\mathsf{b}}\big), \tag{1.21a}$$

$$\pi^{\mathsf{a}}_{k+1} = \mathcal{C}_{k+1}\big(\pi^{\mathsf{f}}_{k+1}\big). \tag{1.21b}$$

Combining equations (1.20) and (1.21b) yields the recursion

$$\pi^{\mathsf{a}}_{k+1} = \mathcal{C}_{k+1} \circ \mathcal{P}_k\big(\pi^{\mathsf{a}}_k\big). \tag{1.22}$$

Using the initial relationship provided by equation (1.21a) and the recursion provided by equation (1.22), we conclude that the filtering density π^{a} exists and only depends on the sequence of observation vectors $\mathbf{y}$ (through the correction operator $\mathcal{C}$). Furthermore, the stability of π^{a} with respect to a variation of $\mathbf{y}$ is a consequence of theorem 2.15 of Law et al. (2015). We conclude theorem 1.1.

Theorem 1.1. *The generic filtering estimation problem, problem 1.1, has a unique and stable solution: it is mathematically well-posed.*

Once the filtering density π^{a} is computed, it is possible to choose an estimate of the unknown truth $\mathbf{x}^{\mathsf{t}}$, written $\mathbf{x}^{\mathsf{a}}$. Different choices are possible. For example, the **mean estimate** is

$$\mathbf{x}^{\mathsf{a}}_k = \int \mathbf{x}_k \, \pi^{\mathsf{a}}_k(\mathbf{x}_k|\mathbf{y}_{k:}) \, \mathrm{d}\mathbf{x}_k, \tag{1.23}$$

where the integral is taken over the whole state space $\mathbb{R}^{N_{\mathsf{x}}}$, and the **maximum *a posteriori*** is

$$\mathbf{x}^{\mathsf{a}}_k = \arg\max_{\mathbf{x}_k} \pi^{\mathsf{a}}_k(\mathbf{x}_k|\mathbf{y}_{k:}), \tag{1.24}$$

where the optimisation is performed over the whole state space $\mathbb{R}^{N_{\mathsf{x}}}$. When π^{a} is Gaussian, both estimates are equal. However, when π^{a} is non-Gaussian, they provide different pieces of information. Suppose for example that π^{a} is bimodal. The maximum *a posteriori* selects one mode, and discards all information about the other mode. By contrast, the mean estimate include information about both modes, but if the modes are distant from each other, it describes a very unlikely estimate of $\mathbf{x}^{\mathsf{t}}$.

Providing an *optimal* filtering estimate $\mathbf{x}^{\mathsf{a}}$ is the primary goal of DA. However, contrary to problem 1.1, the problem which consists in computing *the* filtering estimate $\mathbf{x}^{\mathsf{a}}$ given the sequence of observation vectors $\mathbf{y}$ is not mathematically well-posed. Indeed, the presence of noise corrupts the data and therefore the existence of a solution is not guaranteed. If the solution exists, there is no reason for it to be unique, especially if the observation operator $\mathcal{H}$ is not injective. Finally, even if the solution exists and is unique, there is no reason to think that it is stable with respect to a variation of $\mathbf{y}$.

1.2.2 Terminology for filtering DA problems

In filtering DA, computing of the prediction density π^{f} is called the **forecast step** and computing the filtering density π^{a} is called the **analysis step**, as illustrated by figure 1.3.

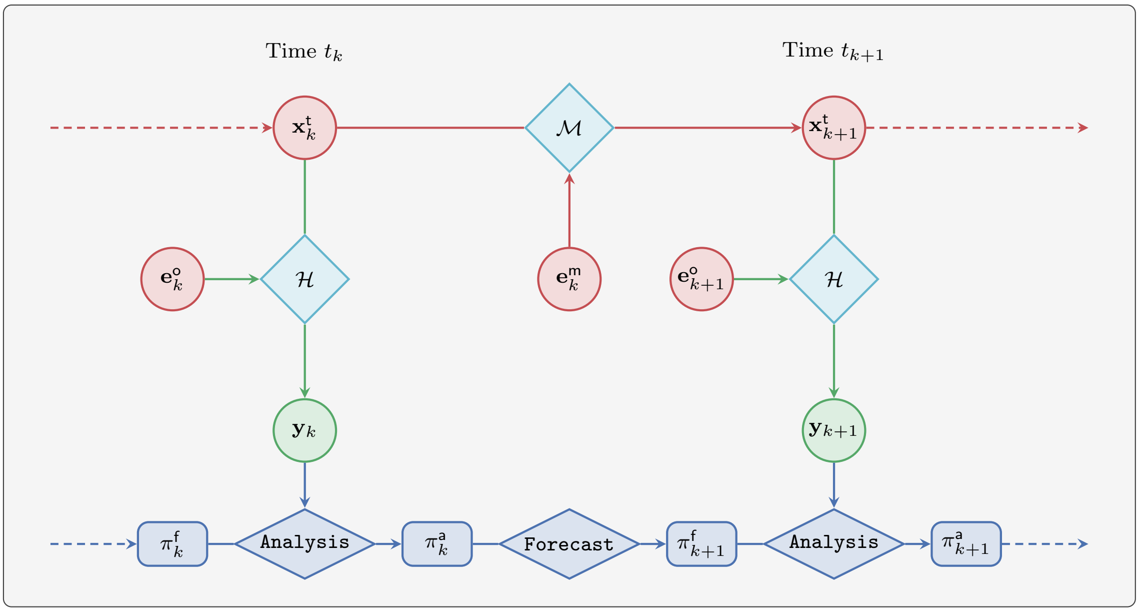

Figure 1.3: Schematic representation of the DA cycles. The hidden part is in red, the observed part in green, the maps are in cyan, and the DA part is in blue.

Therefore, π^{f} and π^{a} may be called the forecast and analysis density, and the filtering estimate $\mathbf{x}^{\mathsf{a}}$ may be called the analysis estimate. The forecast and analysis distributions, whose pdfs are π^{f} and π^{a}, are written ν^{f} and ν^{a}.

1.2.3 Simplified DA systems

In many applications, several assumptions are considered in order to simplify the DA system. In this section, we introduce the most common simplified DA systems.

1.2.3.1 Additive Gaussian and linear system

The DA system is said to be additive Gaussian and linear (GL) if the following conditions are met:

- the initial state vector $\boldsymbol{x}_0$ follows a Gaussian distribution;
- the observation operator $\mathcal{H}$ is a linear application;
- the model $\mathcal{M}$ is a linear application;
- the model and observation errors $\boldsymbol{e}^{\mathsf{m}}$ and $\boldsymbol{e}^{\mathsf{o}}$ follow a centred Gaussian distribution.

Using these hypotheses, the GL system can be written as follows.

GL system

$$\boldsymbol{x}_0 \sim \mathcal{N}\big[\mathbf{x}^{\mathsf{b}}, \mathbf{B}\big], \tag{1.25a}$$

$$\boldsymbol{y}_k = \mathbf{H}\boldsymbol{x}_k + \boldsymbol{e}^{\mathsf{o}}_k, \qquad \boldsymbol{e}^{\mathsf{o}}_k \sim \mathcal{N}\big[\mathbf{0}, \mathbf{R}\big], \tag{1.25b}$$

$$\boldsymbol{x}_{k+1} = \mathbf{M}\boldsymbol{x}_k + \boldsymbol{e}^{\mathsf{m}}_k, \qquad \boldsymbol{e}^{\mathsf{m}}_k \sim \mathcal{N}\big[\mathbf{0}, \mathbf{Q}\big]. \tag{1.25c}$$

The background density π^{b} is the Gaussian density $\mathcal{N}(\mathbf{x}_0|\mathbf{x}^{\mathsf{b}}, \mathbf{B})$. The vector $\mathbf{x}^{\mathsf{b}}$ is the mean of π^{b} and can be used as background estimate. The matrix $\mathbf{B}$ is the background error covariance matrix.[5] The matrices $\mathbf{M}$ and $\mathbf{H}$ are the matrices of $\mathcal{M}$ and $\mathcal{H}$.[6] Finally, $\mathbf{Q}$ and $\mathbf{R}$ are the covariance matrices of the model and observation errors $\boldsymbol{e}^{\mathsf{m}}$ and $\boldsymbol{e}^{\mathsf{o}}$.

The observation and transition densities π^{o} and π^{m}, deduced from equations (1.25b) and (1.25c), are given by

$$\pi^{\mathsf{o}}_k(\mathbf{y}_k|\mathbf{x}_k) = \mathcal{N}(\mathbf{y}_k|\mathbf{H}\mathbf{x}_k, \mathbf{R}), \tag{1.26}$$

$$\pi^{\mathsf{m}}_k(\mathbf{x}_{k+1}|\mathbf{x}_k) = \mathcal{N}(\mathbf{x}_{k+1}|\mathbf{M}\mathbf{x}_k, \mathbf{Q}), \tag{1.27}$$

Finally, the dedicated estimation problem is formalised in problem 1.2.

Problem 1.2 (GL filtering estimation problem)**.** *Given the sequence of observation vectors* $\mathbf{y}$*, compute the analysis density* π^{a} *of the GL system.*

Remark 4. In the system (1.25), the model and observation error covariance matrices $\mathbf{Q}$ and $\mathbf{R}$ do not depend on the time index k. Again, the generalisation to time-dependent $\mathbf{Q}_k$ and $\mathbf{R}_k$ is straightforward in this thesis.

1.2.3.2 Single analysis step of the GL system

When the focus is on a single analysis (SA) step, there is no time evolution. As a consequence, the SA–GL system is simply written as follows.

SA–GL system

$$\boldsymbol{x} \sim \mathcal{N}\big[\mathbf{x}^{\mathsf{b}}, \mathbf{B}\big], \tag{1.28a}$$

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{e}^{\mathsf{o}}, \qquad \boldsymbol{e}^{\mathsf{o}} \sim \mathcal{N}\big[\mathbf{0}, \mathbf{R}\big]. \tag{1.28b}$$

The background density $\pi^{\mathsf{b}} \triangleq \pi[\boldsymbol{x}]$ is the Gaussian density $\mathcal{N}(\mathbf{x}|\mathbf{x}^{\mathsf{b}}, \mathbf{B})$. Using Bayes' theorem, the analysis density $\pi^{\mathsf{a}} \triangleq \pi[\boldsymbol{x}|\boldsymbol{y}]$ is equal to

$$\pi^{\mathsf{a}}(\mathbf{x}|\mathbf{y}) = \frac{\pi^{\mathsf{o}}(\mathbf{y}|\mathbf{x})\,\pi^{\mathsf{b}}(\mathbf{x})}{\pi[\boldsymbol{y}](\mathbf{y})}, \tag{1.29}$$

[5] For simplicity, it is assumed in this thesis that all covariance matrices are symmetric and positive-definite.

[6] Unless specified otherwise, the matrix of a linear map $\mathcal{F} \in \mathbb{F}^{n \leftarrow p}$ is defined in this thesis as the matrix $\mathbf{F} \in \mathbb{R}^{p\times n}$ which represents $\mathcal{F}$ in the canonical bases of $\mathbb{R}^n$ and $\mathbb{R}^p$.

where the observation density $\pi^{\mathsf{o}} \triangleq \pi[\boldsymbol{y}|\boldsymbol{x}]$, deduced from equation (1.28b), is given by

$$\pi^{\mathsf{o}}(\mathbf{y}|\mathbf{x}) = \mathcal{N}(\mathbf{y}|\mathbf{H}\mathbf{x}, \mathbf{R}). \tag{1.30}$$

Finally, the dedicated estimation problem is formalised in problem 1.3.

Problem 1.3 (SA–GL filtering estimation problem)**.** *Given the observation vector* $\mathbf{y}$*, compute the analysis density* π^{a} *of the SA–GL system.*

1.3 Introduction to statistical data assimilation

In this section, we present the principle of statistical DA methods in the context of the SA–GL system, and we derive the so-called best linear unbiased estimate (BLUE) analysis.

1.3.1 Principle of statistical methods

The background error $\boldsymbol{e}^{\mathsf{b}}$ is defined as the random vector

$$\boldsymbol{e}^{\mathsf{b}} \triangleq \boldsymbol{x} - \mathbf{x}^{\mathsf{b}}. \tag{1.31}$$

Because the state vector $\boldsymbol{x}$ follows the Gaussian distribution $\mathcal{N}\big[\mathbf{x}^{\mathsf{b}}, \mathbf{B}\big]$, the expected value of $\boldsymbol{e}^{\mathsf{b}}$ is

$$\mathbb{E}\big[\boldsymbol{e}^{\mathsf{b}}\big] = \mathbb{E}\big[\boldsymbol{x}\big] - \mathbf{x}^{\mathsf{b}} = \mathbf{0}. \tag{1.32}$$

The background estimate $\mathbf{x}^{\mathsf{b}}$ is said to be **unbiased**.

In statistical DA methods, the goal is to provide an optimal unbiased analysis estimate $\mathbf{x}^{\mathsf{a}}$ of the truth. Let $\boldsymbol{e}^{\mathsf{a}}$ be the associated analysis error, defined as

$$\boldsymbol{e}^{\mathsf{a}} \triangleq \boldsymbol{x} - \mathbf{x}^{\mathsf{a}}, \tag{1.33}$$

and let $\mathbf{P}^{\mathsf{a}}$ be the analysis error covariance matrix, defined as

$$\mathbf{P}^{\mathsf{a}} \triangleq \mathbb{V}\big[\boldsymbol{e}^{\mathsf{a}}\big]. \tag{1.34}$$

Here, *optimality* refer to some kind of variance minimisation. For example, the optimal unbiased analysis estimate $\mathbf{x}^{\mathsf{a}}$ could be the one which minimises the trace of $\mathbf{P}^{\mathsf{a}}$. This is formalised in problem 1.4.

Problem 1.4 (Statistical analysis)**.** *Given the observation vector* $\mathbf{y}$*, compute the unbiased analysis estimate* $\mathbf{x}^{\mathsf{a}}$ *of the SA–GL system which minimises the trace of the analysis error covariance matrix* $\mathbf{P}^{\mathsf{a}}$*.*

1.3.2 The BLUE analysis

Suppose now that the analysis estimate $\mathbf{x}^{\mathsf{a}}$ is a linear combination of the background estimate $\mathbf{x}^{\mathsf{b}}$ and of the observation vector $\mathbf{y}$, given by

$$\mathbf{x}^{\mathsf{a}} = \big(\mathbf{I} - \mathbf{K}\mathbf{H}\big)\,\mathbf{x}^{\mathsf{b}} + \mathbf{K}\mathbf{y}, \tag{1.35}$$

where $\mathbf{K}$ is called the **gain** matrix. Equation (1.28b) shows that the expected value of $\boldsymbol{e}^{\mathrm{o}}$ is zero. Furthermore, since $\mathbf{x}^{\mathsf{b}}$ is unbiased, equation (1.35) ensures that $\mathbf{x}^{\mathsf{a}}$ is unbiased as well. Simple linear algebra shows that the associated analysis error covariance matrix $\mathbf{P}^{\mathsf{a}}$ is

$$\mathbf{P}^{\mathsf{a}} = \left(\mathbf{I} - \mathbf{KH}\right)\mathbf{B}\left(\mathbf{I} - \mathbf{KH}\right)^{\mathsf{T}} + \mathbf{KRK}^{\mathsf{T}}. \tag{1.36}$$

Finally, the optimality condition on the trace of $\mathbf{P}^{\mathsf{a}}$ yields the optimal gain matrix, also called Kalman gain matrix, given by

$$\mathbf{K} = \mathbf{BH}^{\mathsf{T}}\left(\mathbf{HBH}^{\mathsf{T}} + \mathbf{R}\right)^{-1}. \tag{1.37}$$

The resulting analysis is called the BLUE analysis and is written, in a more concise but equivalent form, as

$$\mathbf{K} = \mathbf{BH}^{\mathsf{T}}\left(\mathbf{HBH}^{\mathsf{T}} + \mathbf{R}\right)^{-1}, \tag{1.38a}$$

$$\mathbf{x}^{\mathsf{a}} = \mathbf{x}^{\mathsf{b}} + \mathbf{K}\left(\mathbf{y} - \mathbf{Hx}^{\mathsf{b}}\right), \tag{1.38b}$$

$$\mathbf{P}^{\mathsf{a}} = \left(\mathbf{I} - \mathbf{KH}\right)\mathbf{B}. \tag{1.38c}$$

Finally, the following alternate expressions for $\mathbf{K}$ and $\mathbf{P}^{\mathsf{a}}$ are derived using the Sherman–Morrison–Woodbury matrix identity, also known as the Woodbury formula:

$$\mathbf{K} = \left(\mathbf{B}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}, \tag{1.39}$$

$$\mathbf{P}^{\mathsf{a}} = \left(\mathbf{B}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}. \tag{1.40}$$

The BLUE analysis is one of the simplest DA method, yet its implementation can be difficult for several reasons:

- the background and analysis error covariance matrices $\mathbf{B}$ and $\mathbf{P}^{\mathsf{a}}$ have size $N_{\mathrm{x}} \times N_{\mathrm{x}}$, which is hardly storable for high-dimensional systems;
- the computation of the analysis error covariance matrix $\mathbf{P}^{\mathsf{a}}$ requires the inversion of a matrix of size $N_{\mathrm{y}} \times N_{\mathrm{y}}$ or $N_{\mathrm{x}} \times N_{\mathrm{x}}$, depending on whether equation (1.38a) or (1.39) is used for the Kalman gain matrix $\mathbf{K}$;
- the exact background- and observation error covariance matrices $\mathbf{B}$ and $\mathbf{R}$ might be unknown.

Remark 5. The matrices $\mathbf{B}$ and $\mathbf{R}$ are symmetric and positive-definite. As a consequence, both $\mathbf{HBH}^{\mathsf{T}} + \mathbf{R}$ and $\mathbf{B}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ are symmetric and positive-definite hence invertible, and the Kalman gain matrix $\mathbf{K}$, as given by equation (1.38a) or equivalently by equation (1.39), is correctly defined.

1.4 Introduction to variational data assimilation

In this section, we present the principle of variational DA methods as a counterpart to statistical DA methods in the same context, *i.e.*, the SA–GL system. The 3D–Var analysis is then introduced and shown to be equivalent to the BLUE analysis.

1.4.1 Principle of variational methods

In variational DA methods, optimality is measured by the means of a cost function $\mathcal{J}$, such as

$$\mathcal{J}(\mathbf{x}) \triangleq \frac{1}{2}\big(\mathbf{x}-\mathbf{x}^{\mathsf{b}}\big)^{\mathsf{T}}\mathbf{B}^{-1}\big(\mathbf{x}-\mathbf{x}^{\mathsf{b}}\big)+\frac{1}{2}\big(\mathbf{y}-\mathbf{H}\mathbf{x}\big)^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y}-\mathbf{H}\mathbf{x}\big). \tag{1.41}$$

In this definition, the first term measures the departure from the background estimate $\mathbf{x}^{\mathsf{b}}$, while the second term measures the departure from the observation vector $\mathbf{y}$. Both terms are pondered by their relative confidence. This rationale is formalised in problem 1.5.

Problem 1.5 (Variational analysis). *Given the observation vector* $\mathbf{y}$*, compute the analysis estimate* $\mathbf{x}^{\mathsf{a}}$ *of the SA–GL system which minimises the cost function* $\mathcal{J}$*.*

1.4.2 The 3D–Var analysis and its equivalence with the BLUE analysis

The cost function $\mathcal{J}$ is quadratic because the observation operator $\mathbf{H}$ is linear. Moreover, $\mathcal{J}$ is strictly convex because the background and observation error covariance matrices $\mathbf{B}$ and $\mathbf{R}$ are symmetric and positive-definite. Therefore $\mathcal{J}$ has a unique minimiser, which can be obtained by nullifying its gradient

$$\nabla\mathcal{J}|_{\mathbf{x}} = \mathbf{B}^{-1}\big(\mathbf{x}-\mathbf{x}^{\mathsf{b}}\big)-\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y}-\mathbf{H}\mathbf{x}\big). \tag{1.42}$$

The minimiser is given by

$$\mathbf{x}^{\mathsf{a}} = \mathbf{x}^{\mathsf{b}}+\big(\mathbf{B}^{-1}+\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\big)^{-1}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y}-\mathbf{H}\mathbf{x}^{\mathsf{b}}\big), \tag{1.43}$$

and the Hessian matrix of $\mathcal{J}$ at the minimum is given by

$$\operatorname{Hess}\mathcal{J}|_{\mathbf{x}^{\mathsf{a}}} = \mathbf{B}^{-1}+\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}. \tag{1.44}$$

This analysis is called 3D–Var. Using the alternate expressions of the BLUE analysis, equations (1.39) and (1.40), we recognise that the minimiser of the cost function $\mathcal{J}$ is the analysis estimate of the BLUE analysis and that the Hessian matrix of $\mathcal{J}$ at the minimum is the inverse of the analysis error covariance matrix $\mathbf{P}^{\mathsf{a}}$ of the BLUE analysis. This shows that the solutions to problem 1.5 and to problem 1.4 are equal, and that the BLUE and the 3D–Var analyses are two facets of a same solution for the SA–GL system. However, the route to this solution is different. The BLUE analysis requires the inversion of a potentially large matrix, while the 3D–Var analysis requires the minimisation of a cost function, for which one benefits from the long experience of numerical optimisation.

1.5 Properties of the BLUE and 3D–Var analyses

To conclude this introduction chapter, we present basic properties of the BLUE and 3D–Var analyses.

1.5.1 Connection to the filtering estimation problem

In the SA–GL system, both the background and the observation densities π^{b} and π^{o} are Gaussian. This means that the analysis density π^{a}, given by equation (1.29), is the Gaussian density $\mathcal{N}(\mathbf{x}|\mathbf{x}^{\mathsf{a}}, \mathbf{P}^{\mathsf{a}})$, where $\mathbf{P}^{\mathsf{a}}$ and $\mathbf{x}^{\mathsf{a}}$ are given by

$$\mathbf{P}^{\mathsf{a}} = \left(\mathbf{B}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\right)^{-1}, \tag{1.45}$$

$$\mathbf{x}^{\mathsf{a}} = \mathbf{P}^{\mathsf{a}}\left(\mathbf{B}^{-1}\mathbf{x}^{\mathsf{b}} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{y}\right). \tag{1.46}$$

Using the alternate equations (1.39) and (1.40), we recognise here the analysis estimate $\mathbf{x}^{\mathsf{a}}$ and the analysis error covariance matrix $\mathbf{P}^{\mathsf{a}}$ of the BLUE and 3D–Var analyses. This means that the BLUE and 3D–Var analyses are indeed solution to the SA–GL filtering estimation problem, problem 1.3. This result is formalised in theorem 1.2.

Theorem 1.2. *The analysis density π^{a} of the SA–GL system, solution to problem 1.3, is the Gaussian density $\mathcal{N}(\mathbf{x}|\mathbf{x}^{\mathsf{a}}, \mathbf{P}^{\mathsf{a}})$, where $\mathbf{x}^{\mathsf{a}}$ and $\mathbf{P}^{\mathsf{a}}$ are the analysis estimate and the analysis error covariance matrix of the BLUE and 3D–Var analyses, as obtained by equations* (1.38a)*–*(1.38c)*.*

1.5.2 Cycling the BLUE and the 3D–Var analyses

Until now, the BLUE and 3D–Var analyses describe how to perform only one analysis step. The way to implement them in a cycled DA system such as the GL system, is to construct four sequences: $\left(\mathbf{x}^{\mathsf{f}}_k\right)_{k\in\mathbb{N}}$, $\left(\mathbf{x}^{\mathsf{a}}_k\right)_{k\in\mathbb{N}}$, $\left(\mathbf{P}^{\mathsf{f}}_k\right)_{k\in\mathbb{N}}$, and $\left(\mathbf{P}^{\mathsf{a}}_k\right)_{k\in\mathbb{N}}$.

The forecast estimate and error covariance matrix $\mathbf{x}^{\mathsf{f}}$ and $\mathbf{P}^{\mathsf{f}}$ are initialised, using the background estimate and error covariance matrix $\mathbf{x}^{\mathsf{b}}$ and $\mathbf{B}$, as

$$\mathbf{x}^{\mathsf{f}}_0 = \mathbf{x}^{\mathsf{b}}, \tag{1.47a}$$

$$\mathbf{P}^{\mathsf{f}}_0 = \mathbf{B}. \tag{1.47b}$$

Then, using equations (1.38a)–(1.38c), the analysis estimate and error covariance matrix, $\mathbf{x}^{\mathsf{a}}$ and $\mathbf{P}^{\mathsf{a}}$, are obtained from the forecast estimate and error covariance matrix, $\mathbf{x}^{\mathsf{f}}$ and $\mathbf{P}^{\mathsf{f}}$, as

$$\mathbf{K}_k = \mathbf{P}^{\mathsf{f}}_k\mathbf{H}^{\mathsf{T}}\left(\mathbf{H}\mathbf{P}^{\mathsf{f}}_k\mathbf{H}^{\mathsf{T}} + \mathbf{R}\right)^{-1}, \tag{1.48a}$$

$$\mathbf{x}^{\mathsf{a}}_k = \mathbf{x}^{\mathsf{f}}_k + \mathbf{K}_k\left(\mathbf{y}_k - \mathbf{H}\mathbf{x}^{\mathsf{f}}_k\right), \tag{1.48b}$$

$$\mathbf{P}^{\mathsf{a}}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{H}\right)\mathbf{P}^{\mathsf{f}}_k. \tag{1.48c}$$

The only missing is the way to obtain $\mathbf{x}^{\mathsf{f}}_{k+1}$ and $\mathbf{P}^{\mathsf{f}}_{k+1}$ from $\mathbf{x}^{\mathsf{a}}_k$ and $\mathbf{P}^{\mathsf{a}}_k$. The most obvious recursion relationship for $\mathbf{x}^{\mathsf{f}}_{k+1}$ is

$$\mathbf{x}^{\mathsf{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathsf{a}}_k, \tag{1.49}$$

where the dynamical model $\mathbf{M}$ is simply applied to $\mathbf{x}^{\mathsf{a}}_k$. Finally, the simplest recursion relationship for $\mathbf{P}^{\mathsf{f}}_{k+1}$ is

$$\mathbf{P}^{\mathsf{f}}_{k+1} = \mathbf{B}. \tag{1.50}$$

In other words $\mathbf{P}^{\mathsf{f}}$ does not evolve in time. In this case, the recurrence relationship is

Algorithm 1.1: Full assimilation cycle for the cycled BLUE and 3D–Var analyses in the context of the GL system.

```
Input: $\mathbf{x}^{\mathsf{a}}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\mathbf{M}$, $\mathbf{H}$, $\mathbf{B}$, $\mathbf{R}$
$\mathbf{K} = \mathbf{B}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}$        // precomputed
$\mathbf{P}^{\mathsf{a}} = \big(\mathbf{I} - \mathbf{K}\mathbf{H}\big)\mathbf{B}$        // precomputed
Forecast
    $\mathbf{x}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{x}^{\mathsf{a}}$
Analysis
    $\mathbf{x}^{\mathsf{a}} \leftarrow \mathbf{x}^{\mathsf{f}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathsf{f}}\big)$
Output: $\mathbf{x}^{\mathsf{a}}$, $\mathbf{P}^{\mathsf{a}}\ [t_{k+1}]$
```

simplified because $\mathbf{P}^{\mathsf{f}}$ and $\mathbf{P}^{\mathsf{a}}$, as well as $\mathbf{K}$ do not evolve in time. Using the DA terminology, the analysis step is the implementation of equations (1.48a)–(1.48c) and the forecast step is the implementation of equations (1.49) and (1.50). A full cycle (a forecast followed by an analysis) for these cycled BLUE and 3D–Var analyses is described in algorithm 1.1.

This kind of analysis has been used for NWP in the twentieth century by operational centres, first under the form of the BLUE analysis, then, in the 1990s, under the form of the 3D–Var analysis. In the 2000s, the 3D–Var analysis has been replaced in most centres by the 4D–Var analysis, which is a generalisation of the 3D–Var analysis especially suited for smoothing, asynchronous problems. More recently, operational centres are moving toward hybrid approaches which combine the advantage of both the statistical- and the variational approach.

1.5.3 How to deal with nonlinearity and non-Gaussianity

The question of linearity and Gaussianity is relevant, because both ingredients are crucial in the derivation of the BLUE and the 3D–Var analyses and of theorem 1.2, whereas realistic geophysical models are in general nonlinear and error distributions are in general non-Gaussian. Solutions to handle nonlinearity and non-Gaussianity are presented here. However, one must keep in mind that in this case, the BLUE and the 3D–Var analyses may result in different values for the analysis estimate and error covariance matrix $\mathbf{x}^{\mathsf{a}}$ and $\mathbf{P}^{\mathsf{a}}$, and that none of them describes the (non-Gaussian) analysis density π^{a}.

1.5.3.1 Nonlinearity of the observation operator

The generalisation of the 3D–Var analysis to the case of a nonlinear observation operator $\mathcal{H}$ is straightforward. In that case, the cost function to minimise is

$$\mathcal{J}(\mathbf{x}) = \frac{1}{2}\big(\mathbf{x} - \mathbf{x}^{\mathsf{b}}\big)^{\mathsf{T}}\mathbf{B}^{-1}\big(\mathbf{x} - \mathbf{x}^{\mathsf{b}}\big) + \frac{1}{2}\big(\mathbf{y} - \mathcal{H}(\mathbf{x})\big)^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y} - \mathcal{H}(\mathbf{x})\big), \tag{1.51}$$

whose gradient and Hessian matrix are given by

$$\nabla\mathcal{J}|_{\mathbf{x}} = \mathbf{B}^{-1}\big(\mathbf{x} - \mathbf{x}^{\mathsf{b}}\big) - \mathbf{H}_{\mathbf{x}}^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y} - \mathcal{H}(\mathbf{x})\big), \tag{1.52}$$

$$\operatorname{Hess}\mathcal{J}|_{\mathbf{x}} = \mathbf{B}^{-1} + \mathbf{H}_{\mathbf{x}}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}_{\mathbf{x}}, \tag{1.53}$$

where $\mathbf{H}_{\mathbf{x}}$ is the matrix of the differential of $\mathcal{H}$, evaluated at $\mathbf{x}$, commonly known in the DA literature as the **tangent linear** of $\mathcal{H}$. In this case however, the cost function $\mathcal{J}$ may not be quadratic and convex, which means that the existence and uniqueness of its minimum is not guaranteed.

By contrast, the BLUE analysis cannot be generalised to the case of a nonlinear observation operator without an approximation: it requires the explicit linearisation of $\mathcal{H}$ about the background estimate.

1.5.3.2 Nonlinearity of the dynamical model

In the cycled BLUE and 3D–Var analyses, the forecast step consists in equation (1.49), which describes how to obtain $\mathbf{x}^{\mathsf{f}}_{k+1}$ from $\mathbf{x}^{\mathsf{a}}_{k}$. The generalisation of equation (1.49) to a nonlinear dynamical model $\mathcal{M}$ is straightforward:

$$\mathbf{x}^{\mathsf{f}}_{k+1} = \mathcal{M}(\mathbf{x}^{\mathsf{a}}_{k}). \tag{1.54}$$

1.5.3.3 Non-Gaussianity of the error distributions

In the linear and Gaussian case, the cost function of the 3D–Var analysis, given by equation (1.41), is equal to

$$\mathcal{J}(\mathbf{x}) = -\ln\pi^{\mathsf{b}}(\mathbf{x}) - \ln\pi^{\mathsf{o}}(\mathbf{y}|\mathbf{x}), \tag{1.55}$$

$$= -\ln\pi^{\mathsf{a}}(\mathbf{x}) - \ln\pi[\boldsymbol{y}](\mathbf{y}), \tag{1.56}$$

where the additive constant $\ln\pi[\boldsymbol{y}](\mathbf{y})$ only depends on the observation vector $\mathbf{y}$. The generalisation of the 3D–Var analysis to the case of non-Gaussian background and observation densities π^{b} and π^{o} is straightforward. In that case, the cost function to minimise is given by equation (1.55). The minimisation method to use will of course depend on the specific expression of $\mathcal{J}$.

By contrast, in order to define the BLUE analysis in the case of non-Gaussian π^{b} and π^{o}, the vector $\mathbf{x}^{\mathsf{b}}$, and the matrices $\mathbf{B}$ and $\mathbf{R}$ must be defined as follows.

- The vector $\mathbf{x}^{\mathsf{b}}$ is provided as a background estimate. This can be the mean background estimate (the mean of π^{b}), the maximum *a priori* (the mode of π^{b}) or any other vector.

- The background error covariance matrix $\mathbf{B}$ is defined as the covariance matrix of the background error $\boldsymbol{e}^{\mathsf{b}} = \boldsymbol{x} - \mathbf{x}^{\mathsf{b}}$.
- The observation error covariance matrix $\mathbf{R}$ is defined as the covariance matrix of the observation error $\boldsymbol{e}^{\mathsf{o}} = \boldsymbol{y} - \mathbf{H}\boldsymbol{x}$.

Remark 6. When the observation operator is nonlinear but the background and observation densities are Gaussian, equations (1.55) and (1.51) are equivalent. This means that the cost function $\mathcal{J}$ defined by equation (1.55) also covers the case of a nonlinear observation operator $\mathcal{H}$.

2 The ensemble Kalman filter

Contents

The Kalman filter (KF), as introduced by Kalman (1960) is one of the most famous filtering DA algorithm, which allows to recursively compute the first two statistical moments of the forecast and analysis (state) density of the GL system. However, its implementation is impossible in high-dimensional systems, which is the main motivation for the use of approximate methods, such as the singular evolutive extended Kalman (SEEK) filter or the ensemble Kalman filter (EnKF). In the SEEK filter (Pham et al. 1998; Brasseur and Verron 2006), the exact KF algorithm is approximated using explicit low rank factorisation of the error covariance matrices. By contrast, in the EnKF, the exact KF algorithm is approximated by means of an ensemble of state vectors. The original EnKF algorithm has been introduced by Evensen (1994) and amended by Burgers et al. (1998) and Houtekamer and Mitchell (1998), and since then, it has inspired many variants based on similar principles, but whose algorithmic implementation differ.

Section 2.1 is dedicated to the description of the KF algorithm. Section 2.2 presents the EnKF. The consistency and the convergence of the EnKF is studied in section 2.3. Section 2.4 tackles the issue of nonlinearity and non-Gaussianity in the EnKF. Finally, section 2.5 introduces inflation and localisation as means to counteract sampling errors in the EnKF. In this chapter, unless specified otherwise, the DA system is the GL system.

2.1 The Kalman filter

We start this chapter with a presentation of the KF algorithm and of its connection to the GL filtering estimation problem.

2.1.1 The Kalman filter algorithm

In the KF algorithm, the goal is, as in subsection 1.5.2, to recursively construct the four sequences $\big(\mathbf{x}^{\mathsf{f}}_k\big)_{k\in\mathbb{N}}$, $\big(\mathbf{x}^{\mathsf{a}}_k\big)_{k\in\mathbb{N}}$, $\big(\mathbf{P}^{\mathsf{f}}_k\big)_{k\in\mathbb{N}}$, and $\big(\mathbf{P}^{\mathsf{a}}_k\big)_{k\in\mathbb{N}}$. Again, the forecast estimate and error covariance matrix $\mathbf{x}^{\mathsf{f}}$ and $\mathbf{P}^{\mathsf{f}}$ are initialised, using the background estimate and error covariance matrix $\mathbf{x}^{\mathsf{b}}$ and $\mathbf{B}$, as

$$\mathbf{x}^{\mathsf{f}}_0 = \mathbf{x}^{\mathsf{b}}, \tag{2.1a}$$

$$\mathbf{P}^{\mathsf{f}}_0 = \mathbf{B}. \tag{2.1b}$$

The recursion of the KF algorithm[1] is then given by the analysis step:

$$\mathbf{K}_k = \mathbf{P}^{\mathsf{f}}_k\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{P}^{\mathsf{f}}_k\mathbf{H}^{\mathsf{T}}+\mathbf{R}\big)^{-1}, \tag{2.2a}$$

$$\mathbf{x}^{\mathsf{a}}_k = \mathbf{x}^{\mathsf{f}}_k + \mathbf{K}_k\big(\mathbf{y}_k - \mathbf{H}\mathbf{x}^{\mathsf{f}}_k\big), \tag{2.2b}$$

$$\mathbf{P}^{\mathsf{a}}_k = \big(\mathbf{I}-\mathbf{K}_k\mathbf{H}\big)\mathbf{P}^{\mathsf{f}}_k, \tag{2.2c}$$

and by the forecast step:

$$\mathbf{x}^{\mathsf{f}}_{k+1} = \mathbf{M}\mathbf{x}^{\mathsf{a}}_k, \tag{2.3a}$$

$$\mathbf{P}^{\mathsf{f}}_{k+1} = \mathbf{M}\mathbf{P}^{\mathsf{a}}_k\mathbf{M}^{\mathsf{T}}+\mathbf{Q}. \tag{2.3b}$$

The only difference with the cycled BLUE and 3D–Var analyses presented in section 1.5.2 is that in the KF algorithm, the error covariance matrix $\mathbf{P}$ is propagated by the dynamical model $\mathbf{M}$ during the forecast step. This is an important feature, because the time evolution of $\mathbf{P}$ is taken into account, which improves the quality of the analyses. However, implementing the propagation of $\mathbf{P}$, equation (2.3b), can be difficult because it requires N_{x} propagations by the dynamical model (whose matrix is $\mathbf{M}$) and N_{x} propagations by the **adjoint** dynamical model (whose matrix is $\mathbf{M}^{\mathsf{T}}$). A full cycle for the KF algorithm is described in algorithm 2.1.

[1] Also known as the dynamical Riccati recursion.

Algorithm 2.1: Full assimilation cycle for the KF algorithm in the context of the GL system.

```
Input: $\mathbf{x}^{\mathsf{a}}$, $\mathbf{P}^{\mathsf{a}}$ $[t_k]$, $\mathbf{y}$ $[t_{k+1}]$
Parameters: $\mathbf{M}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$
Forecast
    $\mathbf{x}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{x}^{\mathsf{a}}$
    $\mathbf{P}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{P}^{\mathsf{a}}\mathbf{M}^{\mathsf{T}} + \mathbf{Q}$
Analysis
    $\mathbf{K} \leftarrow \mathbf{P}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{P}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}$
    $\mathbf{x}^{\mathsf{a}} \leftarrow \mathbf{x}^{\mathsf{f}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\mathbf{x}^{\mathsf{f}}\big)$
    $\mathbf{P}^{\mathsf{a}} \leftarrow \big(\mathbf{I} - \mathbf{K}\mathbf{H}\big)\mathbf{P}^{\mathsf{f}}$
Output: $\mathbf{x}^{\mathsf{a}}$, $\mathbf{P}^{\mathsf{a}}$ $[t_{k+1}]$
```

2.1.2 The filtering density of GL system

As shown in subsection 1.5.1, the analysis density π_0^{a} of the GL system at time t_0 is the Gaussian density $\mathcal{N}(\mathbf{x}_0|\mathbf{x}_0^{\mathsf{a}}, \mathbf{P}_0^{\mathsf{a}})$, where $\mathbf{P}_0^{\mathsf{a}}$ and $\mathbf{x}_0^{\mathsf{a}}$ are obtained by the BLUE analysis. Equation (1.27) states that the transition density π_0^{m} between times t_0 and t_1 is the Gaussian density $\mathcal{N}(\mathbf{x}_1|\mathbf{M}\mathbf{x}_0, \mathbf{Q})$. From the Chapman–Kolmogorov equation, equation (1.20), we deduce that the forecast density π_1^{f} at time t_1 is the Gaussian density $\mathcal{N}(\mathbf{x}_1|\mathbf{x}_1^{\mathsf{f}}, \mathbf{P}_1^{\mathsf{f}})$, where $\mathbf{P}_1^{\mathsf{f}}$ and $\mathbf{x}_1^{\mathsf{f}}$ are simply given by

$$\mathbf{x}_1^{\mathsf{f}} = \mathbf{M}\mathbf{x}_0^{\mathsf{a}}, \tag{2.4}$$

$$\mathbf{P}_1^{\mathsf{f}} = \mathbf{M}\mathbf{P}_0^{\mathsf{a}}\mathbf{M}^{\mathsf{T}} + \mathbf{Q}. \tag{2.5}$$

Reasoning by recurrence, we conclude theorem 2.1, which *a posteriori* justifies the forecast step in the recursion of the KF algorithm, equations (2.3a)–(2.3b).

Theorem 2.1. *The analysis density π^{a} of the GL system, solution to problem 1.2, and the associated forecast density π^{f} are the Gaussian densities $\mathcal{N}(\mathbf{x}|\mathbf{x}^{\mathsf{a}}, \mathbf{P}^{\mathsf{a}})$ and $\mathcal{N}(\mathbf{x}|\mathbf{x}^{\mathsf{f}}, \mathbf{P}^{\mathsf{f}})$, where $\mathbf{x}^{\mathsf{a}}$, $\mathbf{P}^{\mathsf{a}}$, $\mathbf{x}^{\mathsf{f}}$, and $\mathbf{P}^{\mathsf{f}}$ are obtained using the recursion of the KF algorithm, equations* (2.2a)–(2.2c) *and* (2.3a)–(2.3b).

2.2 The ensemble Kalman filter

This section presents the main algorithmic ingredients of the EnKF, as well as the two reference variants of the EnKF which are used in this thesis: the stochastic EnKF algorithm and the ensemble transform Kalman filter (ETKF) algorithm.

2.2.1 Matrix notation for ensemble DA

In DA, an **ensemble** E is defined as a collection of N_e vectors with N elements, written $\big\{\mathbf{x}_i \in \mathbb{R}^N, i \in (N_\mathrm{e}:1)\big\}$. The **sample mean** $\bar{\mathbf{x}}$ and **sample covariance** matrix $\bar{\mathbf{P}}$ of E are

$$\bar{\mathbf{x}} \triangleq \frac{1}{N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} \mathbf{x}_i, \tag{2.6a}$$

$$\bar{\mathbf{P}} \triangleq \frac{1}{N_\mathrm{e}-1} \sum_{i=1}^{N_\mathrm{e}} (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^\mathsf{T}. \tag{2.6b}$$

For convenience, we introduce a matrix notation as follows. The matrix $\mathbf{E}$ of the ensemble E is defined as the $N \times N_\mathrm{e}$ matrix whose N_e columns are the ensemble members $\mathbf{x}_{N_\mathrm{e}:1}$. Then, we introduce the **normalised anomaly** or **perturbation** matrix $\mathbf{X}$ of E, defined as

$$\mathbf{X} \triangleq \frac{1}{\sqrt{N_\mathrm{e}-1}} \mathbf{E}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^\mathsf{T}/N_\mathrm{e}\big). \tag{2.7}$$

With this matrix notation, the sample mean and covariance matrix $\bar{\mathbf{x}}$ and $\bar{\mathbf{P}}$ of the ensemble E can be obtained as

$$\bar{\mathbf{x}} = \mathbf{E}\mathbf{1}/N_\mathrm{e}, \tag{2.8a}$$

$$\bar{\mathbf{P}} = \mathbf{X}\mathbf{X}^\mathsf{T}. \tag{2.8b}$$

In this thesis, the term *ensemble* refers, without distinction, to an ensemble E or to its matrix $\mathbf{E}$.

2.2.2 Principle of the EnKF

In the KF algorithm, the knowledge on the system is determined by the current estimate and by the associated error covariance matrix. In the EnKF, the knowledge on the system is determined by an ensemble $\mathsf{E} = \big\{\mathbf{x}_i \in \mathbb{R}^{N_\mathrm{x}}, i \in (N_\mathrm{e}:1)\big\}$. Each ensemble member $\mathbf{x}_i$ represents a possible realisation of the state vector $\boldsymbol{x}$, and the current estimate and the associated error covariance matrix are the sample mean and covariance matrix of E. Therefore, the goal in the EnKF is to recursively construct two sequences: $\big(\mathbf{E}^\mathsf{f}_k\big)_{k\in\mathbb{N}}$ and $(\mathbf{E}^\mathsf{a}_k)_{k\in\mathbb{N}}$.

The initial forecast ensemble $\mathbf{E}^\mathsf{f}_0$ is an idenpendant and identically distributed (iid) sample from the background distribution $\mathcal{N}\big[\mathbf{x}^\mathsf{b}, \mathbf{B}\big]$. The forecast and analysis steps describe how to update the forecast and analysis ensembles $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$, as ensemble counterparts of the forecast and analysis steps of the KF algorithm.

The EnKF forecast step is said to be **consistent** if the sample means and covariance matrices of $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$ are related by

$$\bar{\mathbf{x}}^\mathsf{f}_{k+1} = \mathbf{M}\bar{\mathbf{x}}^\mathsf{a}_k, \tag{2.9a}$$

$$\bar{\mathbf{P}}^\mathsf{f}_{k+1} = \mathbf{M}\bar{\mathbf{P}}^\mathsf{a}_k\mathbf{M}^\mathsf{T} + \mathbf{Q}, \tag{2.9b}$$

and likewise, the EnKF analysis step is said to be consistent if the sample means and covariance matrices of $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$ are related by

$$\begin{aligned}\mathbf{K}_k &= \bar{\mathbf{P}}^\mathsf{f}_k\mathbf{H}^\mathsf{T}\big(\mathbf{H}\bar{\mathbf{P}}^\mathsf{f}_k\mathbf{H}^\mathsf{T}+\mathbf{R}\big)^{-1}, && (2.10a)\\ \bar{\mathbf{x}}^\mathsf{a}_k &= \bar{\mathbf{x}}^\mathsf{f}_k+\mathbf{K}_k\big(\mathbf{y}_k-\mathbf{H}\bar{\mathbf{x}}^\mathsf{f}_k\big), && (2.10b)\\ \bar{\mathbf{P}}^\mathsf{a}_k &= \big(\mathbf{I}-\mathbf{K}_k\mathbf{H}\big)\bar{\mathbf{P}}^\mathsf{f}_k. && (2.10c)\end{aligned}$$

As a matter of fact, there are many different ways to perform consistent ensemble updates. This is why in this thesis, the term EnKF does not refer to a specific algorithm, but rather to a class of algorithms, whose principles are the ones aforementioned. The main distinction between different EnKF algorithms is probably whether they are **stochastic**, meaning that their analysis step relies on perturbed observations, or whether they are **deterministic**, meaning that their analysis step relies on square root formulae.[2]

Remark 7. The sample covariance matrix $\bar{\mathbf{P}}$ of an ensemble $\mathbf{E}$ may not be positive-definite, but only positive semi-definite. However, because $\mathbf{R}$ is positive-definite, the matrix $\mathbf{H}\bar{\mathbf{P}}\mathbf{H}^\mathsf{T}+\mathbf{R}$ remains positive-definite. Therefore, the Kalman gain matrix $\mathbf{K}$, as given by equation (2.10a), is correctly defined. Obviously, it may differ from the Kalman gain matrix $\mathbf{K}$ of the KF algorithm, as given by equation (2.2a).

2.2.3 The forecast step of stochastic EnKF algorithms

In stochastic EnKF algorithms, the goal of the forecast step if to independently update each ensemble member using the transition equation of the GL system, equation (1.25c). More precisely, at time t_k, the i-th ensemble member is updated as[3]

$$\mathbf{x}^\mathsf{f}_i(k+1)=\mathbf{M}\mathbf{x}^\mathsf{a}_i(k)+\mathbf{e}^\mathsf{m}_i(k), \tag{2.11}$$

where $\mathbf{e}^\mathsf{m}_i$ is a random draw from the model error distribution $\mathcal{N}\big[\mathbf{0},\mathbf{Q}\big]$ (in other words, $\mathbf{e}^\mathsf{m}_i$ is a realisation of the model error $\boldsymbol{e}^\mathsf{m}$). Using the matrix notation, the ensemble update reads

$$\mathbf{E}^\mathsf{f}_{k+1}=\mathbf{M}\mathbf{E}^\mathsf{a}_k+\mathbf{E}^\mathsf{m}_k, \tag{2.12}$$

where $\mathbf{E}^\mathsf{m}$ is the matrix of the ensemble $\big\{\mathbf{e}^\mathsf{m}_i, i\in(N_\mathrm{e}\!:\!1)\big\}$.

2.2.4 The forecast step of deterministic EnKF algorithms

Without model error, the forecast step of deterministic EnKF algorithms is identical to that of stochastic EnKF algorithms, and relies on the dynamical model $\mathbf{M}$. However, the treatment of model error is more complex with deterministic EnKF algorithms. In this subsection, we present the core method described by Raanes et al. (2015). This method is largely inspired from the analysis step of deterministic EnKF algorithms, described in subsection 2.2.6, and hence the forecast step is split into two parts: first the mean update, and then the perturbation update. The mean update relies on the dynamical model $\mathbf{M}$ and

[2] In principle, the distinction should also inform whether the forecast step is stochastic or deterministic. However, the EnKF is often used without model error in which case the forecast step is always deterministic.

[3] In order to avoid double subscripts in this chapter, a functional notation is used instead of a subscript notation for the time indices if necessary.

is similar to the update of $\mathbf{x}^{\mathsf{f}}$ in the forecast step of the KF algorithm, using equation (2.3a). The perturbation update relies on a square root formula to yield the forecast ensemble $\mathbf{E}^{\mathsf{f}}$.

2.2.4.1 The matrix square root

Different choices are possible for the definition of square roots of matrices. In this thesis, we use the following definition. Let $\mathbf{M}$ be a square matrix, diagonalisable with non-negative eigenvalues. We write

$$\mathbf{M} = \mathbf{G}\mathbf{D}\mathbf{G}^{-1}, \tag{2.13}$$

where $\mathbf{G}$, whose columns are the eigenvectors of $\mathbf{M}$, is invertible and where $\mathbf{D}$ is the diagonal matrix containing the eigenvalues of $\mathbf{M}$ in descending order. We define $\mathbf{M}^{1/2}$, the square root of $\mathbf{M}$, as the matrix

$$\mathbf{M}^{1/2} = \mathbf{G}\mathbf{D}^{1/2}\mathbf{G}^{-1}, \tag{2.14}$$

where $\mathbf{D}^{1/2}$ is the diagonal matrix whose elements are the square roots of the elements of $\mathbf{D}$. With this definition, the square root of $\mathbf{M}$ exists, is unique, and satisfies

$$\mathbf{M} = \left(\mathbf{M}^{1/2}\right)^2. \tag{2.15}$$

2.2.4.2 Mean update

The mean update is simply given by equation (2.9a).

2.2.4.3 Perturbation update

Let $\mathbf{X}^{\mathsf{f}}$ and $\mathbf{X}^{\mathsf{a}}$ be the perturbation matrices of $\mathbf{E}^{\mathsf{f}}$ and $\mathbf{E}^{\mathsf{a}}$. In the core method described by Raanes et al. (2015), the perturbation update is implemented as

$$\mathbf{Z} = (\mathbf{M}\mathbf{X}^{\mathsf{a}}_{k})^{+}, \tag{2.16a}$$

$$\mathbf{T}_{\mathrm{e}} = \mathbf{I} + \mathbf{Z}\mathbf{Q}\mathbf{Z}^{\mathsf{T}}, \tag{2.16b}$$

$$\mathbf{X}^{\mathsf{f}}_{k+1} = \mathbf{M}\mathbf{X}^{\mathsf{a}}_{k}(\mathbf{T}_{\mathrm{e}})^{1/2}, \tag{2.16c}$$

where the pseudo-inverse[4] is used for the non-invertible matrix $\mathbf{M}\mathbf{X}^{\mathsf{a}}$. The matrix $\mathbf{Z}\mathbf{Q}\mathbf{Z}^{\mathsf{T}}$ is symmetric and positive semi-definite, hence diagonalisable with non-negative eigenvalues. As a consequence, the square root of the transformation matrix $\mathbf{T}_{\mathrm{e}}$ exists and equation (2.16c) is correctly defined.

2.2.5 The analysis step of stochastic EnKF algorithms

In stochastic EnKF algorithms, the goal of the analysis step is to update each ensemble member in a similar way as $\mathbf{x}^{\mathsf{a}}$ is updated in the analysis step of the KF algorithm, using equation (2.2b). However, as is, the resulting analysis ensemble $\mathbf{E}^{\mathsf{a}}$ would be inconsistent. A solution to recover a consistent analysis step is to use perturbed observations as follows.

[4] Also known as Moore–Penrose inverse.

2.2.5.1 Ensemble update with perturbed observations

At time t_k, the i-th ensemble member is updated as

$$\mathbf{x}_i^{\mathsf{a}}(k) = \mathbf{x}_i^{\mathsf{f}}(k) + \mathbf{K}_k\big(\mathbf{y}_k + \mathbf{e}_i^{\mathsf{o}}(k) - \mathbf{H}\mathbf{x}_i^{\mathsf{f}}(k)\big), \tag{2.17}$$

where the Kalman gain matrix $\mathbf{K}$ is given by equation (2.10a), and where $\mathbf{e}_i^{\mathsf{o}}$ is a random draw from the observation error distribution $\mathcal{N}\big[\mathbf{0}, \mathbf{R}\big]$ (in other words, $\mathbf{e}_i^{\mathsf{o}}$ is a realisation of the observation error $\boldsymbol{e}^{\mathsf{o}}$). Using the matrix notation, the ensemble update is

$$\mathbf{E}_k^{\mathsf{a}} = \mathbf{E}_k^{\mathsf{f}} + \mathbf{K}_k\big(\mathbf{y}_k\mathbf{1}^{\mathsf{T}} + \mathbf{E}_k^{\mathsf{o}} - \mathbf{H}\mathbf{E}_k^{\mathsf{f}}\big), \tag{2.18}$$

where $\mathbf{E}^{\mathsf{o}}$ is the $N_{\mathrm{y}} \times N_{\mathrm{e}}$ matrix of the ensemble $\big\{\mathbf{e}_i^{\mathsf{o}}, i \in (N_{\mathrm{e}} : 1)\big\}$.

2.2.5.2 Kalman gain matrix in observation space

Define the perturbation matrix in observation space $\mathbf{Y}$ as

$$\mathbf{Y}_k \triangleq \mathbf{H}\mathbf{X}_k^{\mathsf{f}}. \tag{2.19}$$

Using this notation, the Kalman gain matrix $\mathbf{K}$, given by equation (2.10a), is equal to

$$\mathbf{K}_k = \mathbf{X}_k^{\mathsf{f}}\mathbf{Y}_k^{\mathsf{T}}\big(\mathbf{Y}_k\mathbf{Y}_k^{\mathsf{T}} + \mathbf{R}\big)^{-1}. \tag{2.20}$$

2.2.5.3 Kalman gain matrix in ensemble space

Using the Sherman–Morrison–Woodbury matrix identity, it can be shown that the Kalman gain matrix $\mathbf{K}$ is also equal to

$$\mathbf{K}_k = \mathbf{X}_k^{\mathsf{f}}\big(\mathbf{I} + \mathbf{Y}_k^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}_k\big)^{-1}\mathbf{Y}_k^{\mathsf{T}}\mathbf{R}^{-1}. \tag{2.21}$$

In equation (2.20), the Kalman gain matrix $\mathbf{K}$ is formulated in observation space, meaning that the matrix to invert has size $N_{\mathrm{y}} \times N_{\mathrm{y}}$. By contrast, in equation (2.21), the Kalman gain matrix $\mathbf{K}$ is formulated in ensemble space, meaning that the matrix to invert now has size $N_{\mathrm{e}} \times N_{\mathrm{e}}$. When the ensemble size N_{e} is considerably smaller than the number of observations N_{y}, this results in a significant gain in algorithmic complexity.

2.2.6 The analysis step of deterministic EnKF algorithms

In deterministic EnKF algorithms, the analysis step is, as the forecast step, split into two parts: first the mean update, and then the perturbation update. The mean update relies on the Kalman gain matrix $\mathbf{K}$ and is similar to the update of $\mathbf{x}^{\mathsf{a}}$ in the analysis step of the KF algorithm, using equation (2.2b). The perturbation update relies on a square root formula to yield the analysis ensemble $\mathbf{E}^{\mathsf{a}}$.

2.2.6.1 Mean update

The mean update is equation (2.10b), where the Kalman gain matrix $\mathbf{K}$, given by equation (2.10a), can be computed either using equation (2.20) or using equation (2.21).

2.2.6.2 Perturbation update in ensemble space

In the ETKF algorithm (Bishop et al. 2001; Hunt et al. 2007), the perturbation update is implemented as

$$\mathbf{T}_{\mathrm{e}} = \mathbf{I} + \mathbf{Y}_k^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}_k, \tag{2.22a}$$

$$\mathbf{X}_k^{\mathsf{a}} = \mathbf{X}_k^{\mathsf{f}}(\mathbf{T}_{\mathrm{e}})^{-1/2}. \tag{2.22b}$$

The matrix $\mathbf{Y}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}$ is symmetric and positive semi-definite, hence diagonalisable with non-negative eigenvalues. As a consequence, the square root of the transformation matrix $\mathbf{T}_{\mathrm{e}}$ exists and equation (2.22b) is correctly defined.

2.2.6.3 Perturbation update in state space

Using the matrix shift lemma (stated in section 6.4.4 of Asch et al. 2016), it can be shown that the perturbation update of the ETKF algorithm, given by equations (2.22a)–(2.22b), is equivalent to

$$\mathbf{T}_{\mathrm{x}} = \mathbf{I} + \bar{\mathbf{P}}_k^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}, \tag{2.23a}$$

$$\mathbf{X}_k^{\mathsf{a}} = (\mathbf{T}_{\mathrm{x}})^{-1/2}\mathbf{X}_k^{\mathsf{f}}. \tag{2.23b}$$

In this case, the matrix $\bar{\mathbf{P}}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ may not be symmetric. However, both matrices $\bar{\mathbf{P}}^{\mathsf{f}}$ and $\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ are symmetric and positive semi-definite. Therefore, corollary 7.6.2 of Horn and Johnson (2012) ensures that the matrix $\bar{\mathbf{P}}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ is diagonalisable with non-negative eigenvalues, which means that the square root of the transformation matrix $\mathbf{T}_{\mathrm{x}}$ exists and that equation (2.23b) is correctly defined.

In equation (2.23a), the transformation matrix $\mathbf{T}_{\mathrm{x}}$ is formulated in state space, meaning that the matrix to invert has size $N_{\mathrm{x}} \times N_{\mathrm{x}}$. By contrast, in equation (2.22a), the transformation matrix $\mathbf{T}_{\mathrm{e}}$ is formulated in ensemble space, meaning that the matrix to invert has size $N_{\mathrm{e}} \times N_{\mathrm{e}}$. When the ensemble size N_{e} is considerably smaller than the number of variables N_{x}, this results in a significant gain in algorithmic complexity.

2.2.7 Summary

Algorithms 2.2 and 2.3 describe a full assimilation cycle for the stochastic EnKF and the ETKF algorithm, which are the reference variants of the stochastic and deterministic EnKF in this thesis. The algorithmic complexity of both algorithms is reported and compared to the algorithmic complexity of the KF algorithm in table 2.1. The benefit of using ensembles is obvious: we do not need to apply the adjoint model $\mathbf{M}^{\mathsf{T}}$, and the overall algorithmic complexity has been reduced. The price to pay for these cheaper algorithms is that the use of a finite ensemble generates potentially large *sampling errors*. As a consequence, the EnKF does not necessarily solve the GL filtering estimation problem, problem 1.2, and several

Algorithm 2.2: Full assimilation cycle for the stochastic EnKF algorithm in the context of the GL system.

```
Input: $\mathbf{E}^{\mathsf{a}}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\mathbf{M}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$
Forecast
    $\mathbf{E}^{\mathsf{m}} \overset{\text{iid}}{\sim} \mathcal{N}\big[\mathbf{0}, \mathbf{Q}\big]$
    $\mathbf{E}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{E}^{\mathsf{a}} + \mathbf{E}^{\mathsf{m}}$
Analysis
    $\mathbf{E}^{\mathsf{o}} \overset{\text{iid}}{\sim} \mathcal{N}\big[\mathbf{0}, \mathbf{R}\big]$
    $\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}} - 1}$
    $\mathbf{Y} \leftarrow \mathbf{H}\mathbf{X}$
    $\mathbf{K} \leftarrow \mathbf{X}\big(\mathbf{I} + \mathbf{Y}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}\big)^{-1}\mathbf{Y}^{\mathsf{T}}\mathbf{R}^{-1}$
    $\mathbf{E}^{\mathsf{a}} \leftarrow \mathbf{E}^{\mathsf{f}} + \mathbf{K}\big(\mathbf{y}\mathbf{1}^{\mathsf{T}} + \mathbf{E}^{\mathsf{o}} - \mathbf{H}\mathbf{E}^{\mathsf{f}}\big)$
Output: $\mathbf{E}^{\mathsf{a}}\ [t_{k+1}]$
```

approximations are required to counteract sampling errors. This later point is discussed in section 2.5.

2.3 Consistency and convergence of the EnKF

The consistency of an EnKF algorithm is an important property, because it ensures that the algorithm is indeed a reduced-rank version of the original KF algorithm. In this section, we first examine the consistency of the EnKF algorithms, and then we discuss their behaviour in the limit of an infinite ensemble $N_{\mathrm{e}} \to \infty$.

2.3.1 Consistency on average of stochastic EnKF algorithms

2.3.1.1 Consistency of the forecast step

In the forecast step of stochastic EnKF algorithms, described by equation (2.12), the matrix $\mathbf{E}^{\mathsf{m}}$ is constructed as an iid sample from the model error distribution $\mathcal{N}\big[\mathbf{0}, \mathbf{Q}\big]$. By the strong law of large numbers, on average over independent random draws of $\mathbf{E}^{\mathsf{m}}$, the sample mean and covariance matrix of $\mathbf{E}^{\mathsf{m}}$ are almost surely equal to $\mathbf{0}$ and $\mathbf{Q}$. Moreover, the random vector $\boldsymbol{e}^{\mathsf{m}}$ does not depend on the analysis ensemble $\mathbf{E}^{\mathsf{a}}$. Again by the strong law of large numbers, on average over independent random draws of $\mathbf{E}^{\mathsf{m}}$, the matrices $\mathbf{E}^{\mathsf{m}}$ and $\mathbf{E}^{\mathsf{a}}$ are almost surely orthogonal.

In brief, on average over independent random draws of $\mathbf{E}^{\mathsf{m}}$, the following three equations

Algorithm 2.3: Full assimilation cycle for the ETKF algorithm in the context of the GL system. The forecast step is performed with the core method.

```
Input: $\mathbf{E}^{\mathsf{a}}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\mathbf{M}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$
Forecast
    $\bar{\mathbf{x}} \leftarrow \mathbf{E}^{\mathsf{a}}\mathbf{1}/N_{\mathrm{e}}$
    $\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{a}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}$
    $\mathbf{Z} \leftarrow (\mathbf{M}\mathbf{X})^{+}$
    $\mathbf{T}_{\mathrm{e}} \leftarrow \mathbf{I} + \mathbf{Z}\mathbf{Q}\mathbf{Z}$
    $\bar{\mathbf{x}}^{\mathsf{f}} \leftarrow \mathbf{M}\bar{\mathbf{x}}$
    $\mathbf{X}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{X}\mathbf{T}_{\mathrm{e}}^{1/2}$
Analysis
    $\mathbf{Y} \leftarrow \mathbf{H}\mathbf{X}^{\mathsf{f}}$
    $\mathbf{T}_{\mathrm{e}} \leftarrow \mathbf{I} + \mathbf{Y}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}$
    $\mathbf{w} \leftarrow \mathbf{T}_{\mathrm{e}}^{-1}\mathbf{Y}^{\mathsf{T}}\mathbf{R}^{-1}\big(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^{\mathsf{f}}\big)$
    $\bar{\mathbf{x}}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{f}} + \mathbf{X}^{\mathsf{f}}\mathbf{w}$
    $\mathbf{X}^{\mathsf{a}} \leftarrow \mathbf{X}^{\mathsf{f}}\mathbf{T}_{\mathrm{e}}^{-1/2}$
    $\mathbf{E}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{a}}\mathbf{1}^{\mathsf{T}} + \sqrt{N_{\mathrm{e}}-1}\mathbf{X}^{\mathsf{a}}$
Output: $\mathbf{E}^{\mathsf{a}}\ [t_{k+1}]$
```

Table 2.1: Comparison of the algorithmic complexity of the KF algorithm (column KF) and of the stochastic EnKF and the ETKF algorithms (column EnKF) In the forecast step, we count the number of applications of the forward and adjoint models $\mathbf{M}$ and $\mathbf{M}^{\mathsf{T}}$. In the analysis step, we report the complexity of the linear algebra operations, which depends in particular on the algorithmic complexity $T_{\mathbf{H}}$ of applying the linear observation operator $\mathbf{H}$ to a vector. For simplicity, it is assumed here that $N_{\mathrm{e}} \leq N_{\mathrm{y}} \leq N_{\mathrm{x}}$.

Assimilation step	KF	EnKF
Forecast	N_{x} applications of $\mathbf{M}$ N_{x} applications of $\mathbf{M}^{\mathsf{T}}$	N_{e} applications of $\mathbf{M}$ —
Analysis	$\mathcal{O}\big(N_{\mathrm{x}}^2 N_{\mathrm{y}}\big)$	$\mathcal{O}\big(T_{\mathbf{H}}N_{\mathrm{e}} + N_{\mathrm{y}}^2 N_{\mathrm{e}} + N_{\mathrm{x}}N_{\mathrm{e}}^2\big)$

almost surely hold:

$$\mathbf{E}^{\mathsf{m}}_k \mathbf{1} = \mathbf{0}, \tag{2.24a}$$

$$\mathbf{X}^{\mathsf{m}}_k (\mathbf{X}^{\mathsf{m}}_k)^{\mathsf{T}} = \mathbf{Q}, \tag{2.24b}$$

$$\mathbf{X}^{\mathsf{a}}_k (\mathbf{X}^{\mathsf{m}}_k)^{\mathsf{T}} = \mathbf{0}, \tag{2.24c}$$

where $\mathbf{X}^{\mathsf{m}}$ is the perturbation matrix of $\mathbf{E}^{\mathsf{m}}$.

Using equations (2.9a)–(2.9b), as well as the equation of the EnKF forecast step, equation (2.12), we directly obtain the consistency relationships equations (2.24a)–(2.24c). Therefore, we conclude theorem 2.2.

Theorem 2.2. *The forecast step of stochastic EnKF algorithms, as described by equation* (2.12)*, is almost surely consistent on average over independent random draws of* $\mathbf{E}^{\mathsf{m}}$*.*

2.3.1.2 Consistency of the analysis step

Using the same reasoning, it is possible to show that using the analysis step of stochastic EnKF algorithms, described by equation (2.18), the equation of the Kalman gain, equation (2.21), and average relationships for $\mathbf{E}^{\mathsf{o}}$, similar to equations (2.9a)–(2.9b), we directly obtain the consistency relationships equations (2.10a)–(2.10c). Therefore, we conclude theorem 2.3.

Theorem 2.3. *The analysis step of stochastic EnKF algorithms, as described by equation* (2.18)*, is almost surely consistent on average over independent random draws of* $\mathbf{E}^{\mathsf{o}}$*.*

2.3.2 Consistency of deterministic EnKF algorithms

The forecast and analysis steps of deterministic EnKF algorithms are split into two parts: the mean update, and the perturbation update. Such an update is consistent if

- the mean update is consistent, in other words it satisfies equation (2.9a) for the forecast step or equation (2.10b) for the analysis step (by construction this is always the case);
- the perturbation update is consistent, in other words it satisfies equation (2.9b) for the forecast step,or equation (2.10c) for the analysis step;
- the perturbation matrix is centred, in other words $\mathbf{X}^{\mathsf{f}}\mathbf{1} = \mathbf{0}$ for the forecast step or $\mathbf{X}^{\mathsf{a}}\mathbf{1} = \mathbf{0}$ for the analysis step.

The last condition is necessary to ensure that the perturbation update does not mess up with the mean update.

2.3.2.1 Consistency of the forecast step

Following Raanes et al. (2015), the forecast perturbation matrix $\mathbf{X}^{\mathsf{f}}$, as defined by equations (2.16a)–(2.16c), satisfies

$$\mathbf{X}^{\mathsf{f}}_{k+1} \big(\mathbf{X}^{\mathsf{f}}_{k+1}\big)^{\mathsf{T}} = \mathbf{M}\bar{\mathbf{P}}^{\mathsf{a}}_k \mathbf{M}^{\mathsf{T}} + \mathbf{\Pi}^{\mathsf{a}}_k \mathbf{Q} \big(\mathbf{\Pi}^{\mathsf{a}}_k\big)^{\mathsf{T}}, \tag{2.25}$$

where $\mathbf{\Pi}^{\mathsf{a}}$ is the projector onto the subspace spanned by $\mathbf{M}\mathbf{X}^{\mathsf{a}}$, obtained as

$$\mathbf{\Pi}^{\mathsf{a}}_{k} = \mathbf{M}\mathbf{X}^{\mathsf{a}}_{k}\big(\mathbf{M}\mathbf{X}^{\mathsf{a}}_{k}\big)^{+}. \tag{2.26}$$

Furthermore, if the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}_{k}$ is centred, then the forecast perturbation matrix $\mathbf{X}^{\mathsf{f}}_{k+1}$ is by construction centred as well.

The quantity $\mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$ is the *two-sided* projection of the model error covariance matrix $\mathbf{Q}$ on the subspace spanned by $\mathbf{M}\mathbf{X}^{\mathsf{a}}$. The **residual** model error is defined as the model error corresponding to the residual model error covariance matrix $\mathbf{Q} - \mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$. By construction, the core method cannot take into account such error. However, if the residual model error is null, then equation (2.25) is equivalent to equation (2.9b). We conclude theorem 2.4.

Theorem 2.4. *The forecast step of the ETKF algorithm, as described by the mean update, equation* (2.9a)*, and the perturbation update, equations* (2.16a)*–*(2.16c)*, is consistent if, and only if the residual model error covariance matrix* $\mathbf{Q} - \mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$*, with* $\mathbf{\Pi}^{\mathsf{a}}$ *being defined by equation* (2.26)*, is null.*

Remark 8. If the rank of $\mathbf{M}\mathbf{X}^{\mathsf{a}}$ is greater than or equal to N_{x}, then the residual model error covariance matrix $\mathbf{Q} - \mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$ is null. This can only happen if $N_{\mathrm{e}} \geq N_{\mathrm{x}}$, which is never the case in realistic applications.

2.3.2.2 Consistency of the analysis step

Using the Sherman–Morrison–Woodbury matrix identity, it is possible to show that the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$, as defined by equations (2.22a)–(2.22b) or equivalently by equations (2.23a)–(2.23b), satisfies

$$\mathbf{X}^{\mathsf{a}}_{k}(\mathbf{X}^{\mathsf{a}}_{k})^{\mathsf{T}} = \big(\mathbf{I} - \mathbf{K}_{k}\mathbf{H}\big)\bar{\mathbf{P}}^{\mathsf{f}}_{k}. \tag{2.27}$$

Furthermore, the forecast perturbation matrix $\mathbf{X}^{\mathsf{f}}$ being assumed centred, the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ is by construction centred. We conclude theorem 2.5.

Theorem 2.5. *The analysis step of the ETKF algorithm, as described by the mean update, equation* (2.10b)*, and the perturbation update, equations* (2.22a)*–*(2.22b) *or equivalently equations* (2.23a)*–*(2.23b)*, is consistent.*

2.3.2.3 Rotation of the ensemble

Let $\mathbf{U}$ be an $N_{\mathrm{e}} \times N_{\mathrm{e}}$ orthogonal matrix such that $\mathbf{U}\mathbf{1} = \mathbf{1}$. The sample mean and covariance matrix of an ensemble $\mathbf{E}$ are the same as the ones of the *rotated* ensemble $\mathbf{E}\mathbf{U}$. This means that the consistency of the forecast and analysis steps are not altered if the output ensemble is rotated. As a consequence, the orthogonal matrix $\mathbf{U}$ can be freely chosen to improve the performances of the algorithm. The orthogonal matrix $\mathbf{U}$ which minimises the displacement between the prior and updated perturbations is $\mathbf{U} = \mathbf{I}$ (Ott et al. 2004). However, it is remarkable that adding random rotations after the analysis step can be beneficial for the performances of deterministic EnKF algorithms, as observed by Sakov and Oke (2008b).

2.3.3 Convergence of stochastic EnKF algorithms

The initial forecast ensemble $\mathbf{E}_0^{\mathsf{f}}$ is constructed as an iid sample from the background distribution $\mathcal{N}\big[\mathbf{x}^{\mathsf{b}}, \mathbf{B}\big]$. By the strong law of large numbers, in the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$, the sample mean and covariance matrix of $\mathbf{E}_0^{\mathsf{f}}$ almost surely converge towards $\mathbf{x}^{\mathsf{b}}$ and $\mathbf{B}$.

Then, for similar reasons as in subsection 2.3.1, in the limit $N_{\mathrm{e}} \to \infty$, the forecast and analysis steps of stochastic EnKF algorithms, described by equations (2.12) and (2.18), are almost surely consistent. However, even though the forecast and analysis steps are individually consistent, we cannot directly conclude the convergence of the sample quantities using the law of large numbers, because the ensemble members are not independent any more. Indeed, during the analysis step of stochastic EnKF algorithms, each analysis ensemble member $\mathbf{x}_i^{\mathsf{a}}$ depends on the Kalman gain matrix $\mathbf{K}$, which is itself computed using the whole forecast ensemble $\mathbf{E}^{\mathsf{f}}$. The interaction between ensemble members is here of type *mean-field interaction.*[5]

Nevertheless, as shown by Le Gland et al. (2011) and Mandel et al. (2011), the sample mean and covariance matrix of the forecast and analysis ensembles obtained using stochastic EnKF algorithms, described by equations (2.12) and (2.18), almost surely converge towards the forecast and analysis estimates and error covariance matrices obtained using recursion of the KF algorithm, equations (2.2a)–(2.2c) and (2.3a)–(2.3b), in the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$, in other words

$$\lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{x}}_k^{\mathsf{f}} = \mathbf{x}_k^{\mathsf{f}}, \qquad \lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{x}}_k^{\mathsf{a}} = \mathbf{x}_k^{\mathsf{a}}, \tag{2.28a}$$

$$\lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{P}}_k^{\mathsf{f}} = \mathbf{P}_k^{\mathsf{f}}, \qquad \lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{P}}_k^{\mathsf{a}} = \mathbf{P}_k^{\mathsf{a}}. \tag{2.28b}$$

This result is formalised in theorem 2.6.

Theorem 2.6. *The sample mean and covariance matrix of the forecast and analysis ensembles obtained using stochastic EnKF algorithms, described by equations* (2.12) *and* (2.18), *converge almost surely towards the forecast and analysis estimates and error covariance matrices obtained using the recursion of the KF algorithm, equations* (2.2a)–(2.2c) *and* (2.3a)–(2.3b), *in the limit of an infinite ensemble,* $N_{\mathrm{e}} \to \infty$*.*

2.3.4 Convergence of deterministic EnKF algorithms

Suppose that the sample mean and covariance matrix of the initial forecast ensemble $\mathbf{E}_0^{\mathsf{f}}$ are exactly the background estimate and error covariance matrix $\mathbf{x}^{\mathsf{b}}$ and $\mathbf{B}$, and that the residual model error covariance matrix $\mathbf{Q} - \mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$ is always null. In this case, a simple induction with the results of theorems 2.4 and 2.5 shows that the sample mean and covariance matrix of the forecast and analysis ensembles $\mathbf{E}^{\mathsf{f}}$ and $\mathbf{E}^{\mathsf{a}}$ obtained using the ETKF algorithm are equal to the forecast and analysis estimates and error covariance matrices obtained using the

[5] Also called *inbreeding* by the historical EnKF community.

recursion of the KF algorithm, equations (2.2a)–(2.2c) and (2.3a)–(2.3b), in other words

$$\bar{\mathbf{x}}^{\mathsf{f}}_k = \mathbf{x}^{\mathsf{f}}_k, \qquad \bar{\mathbf{x}}^{\mathsf{a}}_k = \mathbf{x}^{\mathsf{a}}_k, \tag{2.29a}$$

$$\bar{\mathbf{P}}^{\mathsf{f}}_k = \mathbf{P}^{\mathsf{f}}_k, \qquad \bar{\mathbf{P}}^{\mathsf{a}}_k = \mathbf{P}^{\mathsf{a}}_k. \tag{2.29b}$$

This is formalised by theorem 2.7.

Theorem 2.7. *If the sample mean and covariance matrix of the initial forecast ensemble $\mathbf{E}^{\mathsf{f}}_0$ are exactly the background estimate and error covariance matrix $\mathbf{x}^{\mathsf{b}}$ and $\mathbf{B}$, and if the residual model error covariance matrix $\mathbf{Q} - \mathbf{\Pi}^{\mathsf{a}}\mathbf{Q}\mathbf{\Pi}^{\mathsf{a}}$, with $\mathbf{\Pi}^{\mathsf{a}}$ being defined by equation* (2.26)*, is always null, then the sample mean and covariance matrix of the forecast and analysis ensembles $\mathbf{E}^{\mathsf{f}}$ and $\mathbf{E}^{\mathsf{a}}$ obtained using the ETKF algorithm are equal to the forecast and analysis estimates and error covariance matrices obtained using the recursion of the KF algorithm, equations* (2.2a)–(2.2c) *and* (2.3a)–(2.3b)*.*

The first condition of theorem 2.7 is met, for example, if the initial forecast ensemble $\mathbf{E}^{\mathsf{f}}_0$ is constructed as

$$\mathbf{E}^{\mathsf{f}}_0 = \mathbf{x}^{\mathsf{b}}\mathbf{1}^{\mathsf{T}} + \sqrt{N_{\mathrm{e}} - 1}\mathbf{X}, \tag{2.30}$$

where $\mathbf{X}\mathbf{X}^{\mathsf{T}} = \mathbf{B}$ is a Choleski factorisation of the background error covariance matrix $\mathbf{B}$, and $N_{\mathrm{e}} = N_{\mathrm{x}}$. The second condition of theorem 2.7 is met, for example, if there is no model error.

2.4 Nonlinearity and non-Gaussianity in the EnKF

Linearity and Gaussianity are crucial hypotheses in the derivation of the KF algorithm and of the different variants of the EnKF, which are rarely satisfied in realistic applications. The standard way of dealing with nonlinearity in the KF algorithm is to use the extended Kalman filter (EKF), in which nonlinear functions are linearised about the current estimate. By contrast, nonlinearity can be treated in the EnKF without the need for explicit linearisation, as presented in this section. However, as already stated in subsection 1.5.3, in the nonlinear and non-Gaussian case, the analysis density π^{a} is non-Gaussian and therefore cannot be described by the analysis estimate $\mathbf{x}^{\mathsf{a}}$ and error covariance matrix $\mathbf{P}^{\mathsf{a}}$ obtained with the KF algorithm. Moreover, in the nonlinear and non-Gaussian case, the convergence results of section 2.3 do not necessary hold.

2.4.1 Nonlinearity of the observation operator

2.4.1.1 Generalisation of the EnKF to nonlinear observation operators

A first approach to deal with a nonlinear observation operator $\mathcal{H}$ in the EnKF could be, as for the BLUE analysis, to use the tangent linear of $\mathcal{H}$. However, it is possible to use the ensemble to avoid the potentially costly computation of the tangent linear of $\mathcal{H}$. For example consider the analysis step of stochastic EnKF algorithms, as described in section 2.2.5. The (linear) observation operator $\mathbf{H}$ is used in two different places.

First, $\mathbf{H}$ is used in equation (2.17), which describes the update of each ensemble member. The generalisation to a nonlinear observation operator $\mathcal{H}$ is straightforward here: the i-th ensemble member is updated as

$$\mathbf{x}_i^{\mathsf{a}}(k) = \mathbf{x}_i^{\mathsf{f}}(k) + \mathbf{K}_k\Big[\mathbf{y}_k + \mathbf{e}_i^{\mathsf{o}}(k) - \mathcal{H}\big(\mathbf{x}_i^{\mathsf{f}}(k)\big)\Big]. \tag{2.31}$$

Second, $\mathbf{H}$ is used to compute the perturbation matrix in observation space $\mathbf{Y}$ with equation (2.19). Using the definition of the perturbation matrix $\mathbf{X}$ of an ensemble $\mathbf{E}$, equation (2.7), we obtain the following relationship for $\mathbf{Y}$:

$$\mathbf{Y}_k = \mathbf{H}\mathbf{E}_k^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}} - 1}, \tag{2.32}$$

where the generalisation to a nonlinear observation operator $\mathcal{H}$ is more natural. In that case, $\mathbf{Y}$ is defined by

$$\mathbf{Y}_k \triangleq \mathcal{H}\big(\mathbf{E}_k^{\mathsf{f}}\big)\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}} - 1}, \tag{2.33}$$

where $\mathcal{H}(\mathbf{E})$ is the $N_{\mathrm{y}} \times N_{\mathrm{e}}$ matrix obtained by applying $\mathcal{H}$ column-wise to the ensemble $\mathbf{E}$ (in other words, $\mathcal{H}$ is applied to each ensemble member $\mathbf{x}_i$ of $\mathbf{E}$).

The generalisation of most EnKF algorithms, including the stochastic EnKF algorithm and the ETKF algorithm, to a nonlinear observation operator $\mathcal{H}$ follows these principles. However, it should be clear that, using this approach, the linearisation is implicit.

Indeed, let $\bar{\mathbf{H}}$ be the sample observation operator, defined as

$$\bar{\mathbf{H}}_k \triangleq \mathbf{Y}_k\big(\mathbf{X}_k^{\mathsf{f}}\big)^{+}. \tag{2.34}$$

By construction, $\bar{\mathbf{H}}$ satisfies

$$\bar{\mathbf{P}}_k^{\mathsf{f}}\bar{\mathbf{H}}_k^{\mathsf{T}} = \mathbf{X}_k^{\mathsf{f}}\mathbf{Y}_k^{\mathsf{T}}, \tag{2.35}$$

$$\bar{\mathbf{H}}_k\bar{\mathbf{P}}_k^{\mathsf{f}}\bar{\mathbf{H}}_k^{\mathsf{T}} = \mathbf{Y}_k\mathbf{Y}_k^{\mathsf{T}}. \tag{2.36}$$

In other words, the generalised EnKF analysis step is similar to a (non-generalised) EnKF analysis step, in which the observation operator $\mathbf{H}$ has been replaced by the sample observation operator $\bar{\mathbf{H}}$. Furthermore, if the forecast ensemble $\mathbf{E}^{\mathsf{f}}$ is indeed an iid sample from the forecast distribution ν^{f}, and if this distribution is Gaussian, then it can be shown (see *e.g.*, Raanes et al. 2019b) that, in the limit of an infinite ensemble $N_{\mathrm{e}} \to \infty$, $\bar{\mathbf{H}}$ almost surely converges, and its limit is given by

$$\lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{H}}_0 = \mathbb{E}[\mathbf{H}_{\boldsymbol{x}_0}], \tag{2.37a}$$

$$\lim_{N_{\mathrm{e}}\to\infty} \bar{\mathbf{H}}_{k+1} = \mathbb{E}\big[\mathbf{H}_{\boldsymbol{x}_{k+1}|\boldsymbol{y}_{k:}}\big], \tag{2.37b}$$

where $\mathbf{H}_{\mathbf{x}}$ is the tangent linear of $\mathcal{H}$, evaluated at $\mathbf{x}$. In conclusion, the implicit linearisation is in theory a bit different from the explicit linearisation, but it has the major advantage that the computation of the tangent linear of $\mathcal{H}$ is unnecessary.

2.4.1.2 Variational analysis in the EnKF

Another approach to deal with a nonlinear observation operator $\mathcal{H}$ is to include variational analysis in the EnKF. Using the formalism of the 3D–Var analysis, the cost function to minimise is

$$\mathcal{J}_k(\mathbf{x}_k) = \frac{1}{2}\left(\mathbf{x}_k - \bar{\mathbf{x}}^{\mathsf{f}}_k\right)^{\mathsf{T}}\left(\bar{\mathbf{P}}^{\mathsf{f}}_k\right)^{+}\left(\mathbf{x}_k - \bar{\mathbf{x}}^{\mathsf{f}}_k\right) + \frac{1}{2}\left(\mathbf{y}_k - \mathcal{H}(\mathbf{x}_k)\right)^{\mathsf{T}}\mathbf{R}^{-1}\left(\mathbf{y}_k - \mathcal{H}(\mathbf{x}_k)\right), \tag{2.38}$$

where the pseudo-inverse is used for the non-invertible matrix $\bar{\mathbf{P}}^{\mathsf{f}}$ instead of the inverse. Following Hunt et al. (2007), Bocquet (2011) and Bocquet and Sakov (2013), if the minimisation is performed in the ensemble space, the vector $\mathbf{x}$ can be written

$$\mathbf{x}_k = \bar{\mathbf{x}}^{\mathsf{f}}_k + \mathbf{X}^{\mathsf{f}}_k\mathbf{w}_k, \tag{2.39}$$

where $\mathbf{w}$ is a weight vector. However, the perturbation matrix $\mathbf{X}^{\mathsf{f}}$ is centred, which means that

$$\forall\alpha \in \mathbb{R}, \quad \mathbf{X}^{\mathsf{f}}_k\mathbf{w}_k = \mathbf{X}^{\mathsf{f}}_k(\mathbf{w}_k + \alpha\mathbf{1}). \tag{2.40}$$

A solution to lift the degeneracy of the variational problem is to constraint the solution to the null space of $\mathbf{X}^{\mathsf{f}}$. This can be done by adding a term in the cost function in order to fix the gauge. The regularised cost function, derived in terms of $\mathbf{w}$, reads

$$\mathcal{J}_k(\mathbf{w}_k) = \frac{1}{2}\mathbf{w}_k^{\mathsf{T}}\mathbf{w}_k + \frac{1}{2}\Big[\mathbf{y}_k - \mathcal{H}\big(\bar{\mathbf{x}}^{\mathsf{f}}_k + \mathbf{X}^{\mathsf{f}}_k\mathbf{w}_k\big)\Big]^{\mathsf{T}}\mathbf{R}^{-1}\Big[\mathbf{y}_k - \mathcal{H}\big(\bar{\mathbf{x}}^{\mathsf{f}}_k + \mathbf{X}^{\mathsf{f}}_k\mathbf{w}_k\big)\Big]. \tag{2.41}$$

A common approach is to minimise the cost function $\mathcal{J}$ using an iterative method, for which we need to specify the gradient and the Hessian matrix. The gradient of $\mathcal{J}$ is

$$\nabla\mathcal{J}_k|_{\mathbf{w}_k} = \mathbf{w}_k - \mathbf{Y}_k^{\mathsf{T}}\mathbf{R}^{-1}\Big[\mathbf{y}_k - \mathcal{H}\big(\bar{\mathbf{x}}^{\mathsf{f}}_k + \mathbf{X}^{\mathsf{f}}_k\mathbf{w}_k\big)\Big], \tag{2.42}$$

with $\mathbf{Y} = \mathbf{H}_{\mathbf{x}}\mathbf{X}^{\mathsf{f}}$, where $\mathbf{H}_{\mathbf{x}}$ is the tangent linear of $\mathcal{H}$ computed at $\mathbf{x} = \bar{\mathbf{x}}^{\mathsf{f}} + \mathbf{X}^{\mathsf{f}}\mathbf{w}$, and the Hessian matrix of $\mathcal{J}$ can be approximated by

$$\operatorname{Hess}\mathcal{J}_k|_{\mathbf{w}_k} \approx \mathbf{I} + \mathbf{Y}_k^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{Y}_k. \tag{2.43}$$

Two methods exist in order to avoid the computation of the tangent linear of $\mathcal{H}$: either downscale or transform the forecast perturbations (Sakov et al. 2012).

Finally, once the minimisation problem is solved, the analysis mean and perturbation matrix $\mathbf{x}^{\mathsf{a}}$ and $\mathbf{X}^{\mathsf{a}}$ can be obtained as

$$\bar{\mathbf{x}}^{\mathsf{a}}_k = \bar{\mathbf{x}}^{\mathsf{f}}_k + \mathbf{X}^{\mathsf{f}}_k\mathbf{w}^{\mathsf{a}}_k, \tag{2.44}$$

$$\mathbf{X}^{\mathsf{a}}_k = \mathbf{X}^{\mathsf{f}}_k\big(\operatorname{Hess}\mathcal{J}_k|_{\mathbf{w}^{\mathsf{a}}_k}\big)^{-1/2}, \tag{2.45}$$

where $\mathbf{w}^{\mathsf{a}}$ is the minimiser of the regularised cost function $\mathcal{J}$. The resulting DA algorithm is called the maximum likelihood ensemble filter (MLEF) algorithm, as originally introduced by Zupanski (2005). When the observation operator $\mathcal{H}$ is linear, the MLEF and the ETKF algorithms are equivalent. However, when $\mathcal{H}$ is nonlinear, the MLEF algorithm performs significantly better than the ETKF algorithm, because the nonlinearity of $\mathcal{H}$ is used in a more consistent way (Asch et al. 2016).

2.4.2 Nonlinearity of the dynamical model

During the forecast step of stochastic EnKF algorithms, each ensemble member is independently updated using the transition equation of the system, yielding equation (2.11). The generalisation of equation (2.11) to a nonlinear dynamical model $\mathcal{M}$ is straightforward. In that case, the i-th ensemble member is updated as

$$\mathbf{x}_i^{\mathsf{f}}(k+1) = \mathcal{M}\big(\mathbf{x}_i^{\mathsf{a}}(k)\big) + \mathbf{e}_i^{\mathsf{m}}(k). \tag{2.46}$$

By construction, this corresponds to a Monte Carlo (MC) simulation of the transition equation, which means that, if $\mathbf{E}_k^{\mathsf{a}}$ is distributed according to the (non-Gaussian) analysis distribution ν_k^{a}, then $\mathbf{E}_{k+1}^{\mathsf{f}}$ is distributed according to the (non-Gaussian) forecast distribution ν_{k+1}^{f}.

Finally, the generalisation of the forecast step of deterministic EnKF algorithms to a nonlinear dynamical model $\mathcal{M}$ follow similar principles as the generalisation of the analysis step of EnKF algorithms to a nonlinear observation operator $\mathcal{H}$, as described in the previous subsection.

2.4.3 Non-Gaussianity of the error distributions

When the background, observation, and transition densities π^{b}, π^{o}, and π^{m} are not Gaussian, then the vector $\mathbf{x}^{\mathsf{b}}$ and the matrices $\mathbf{B}$, $\mathbf{R}$, and $\mathbf{Q}$ are defined as follows.

- The vector $\mathbf{x}^{\mathsf{b}}$ is provided as a background estimate.
- The background error covariance matrix $\mathbf{B}$ is defined as the covariance matrix of the background error $\boldsymbol{e}^{\mathsf{b}} = \boldsymbol{x} - \mathbf{x}^{\mathsf{b}}$.
- The observation error covariance matrix $\mathbf{R}$ is defined as the covariance matrix of the observation error $\boldsymbol{e}^{\mathsf{o}} = \boldsymbol{y} - \mathbf{H}\boldsymbol{x}$.
- The model error covariance matrix $\mathbf{Q}_k$ is defined as the covariance matrix of the model error $\boldsymbol{e}_k^{\mathsf{m}} = \boldsymbol{x}_{k+1} - \mathbf{M}\boldsymbol{x}_k$.

2.5 Inflation and localisation

Sampling error is the main difficulty which arises when trying to apply the EnKF to a high-dimensional system. In this section, we first present several manifestations of sampling error, and we introduce inflation and localisation, the two most common techniques in the EnKF designed to counteract sampling errors.

2.5.1 Manifestations of sampling error

In ensemble DA, the term sampling error usually designates all the errors which originate from the use of a finite ensemble.

Suppose that the forecast ensemble $\mathbf{E}^{\mathsf{f}}$ is an iid sample from the forecast distribution $\nu^{\mathsf{f}} = \mathcal{N}\big[\mathbf{x}^{\mathsf{f}}, \mathbf{P}^{\mathsf{f}}\big]$, where $\mathbf{x}^{\mathsf{f}}$ and $\mathbf{P}^{\mathsf{f}}$ are the exact forecast estimate and forecast error covariance matrix, obtained using the dynamical Riccati recursion equations (2.2a)–(2.2c) and (2.3a)–(2.3b). We show two consequences of sampling error in this case.

2.5.1.1 Spurious correlations at long distance

Following Carrassi et al. (2018), and references therein, as long as m and n are two distinct indices in $(N_\mathrm{x}:1)$, the error in the estimation of the forecast error covariance matrix satisfies

$$\mathbb{E}\Big[\big[\bar{\mathbf{P}}^\mathsf{f}_k - \mathbf{P}^\mathsf{f}_k\big]^2_{m,n}\Big] = \frac{1}{N_\mathrm{e}-1}\Big[\big[\mathbf{P}^\mathsf{f}_k\big]^2_{m,n} + \big[\mathbf{P}^\mathsf{f}_k\big]_{m,m}\big[\mathbf{P}^\mathsf{f}_k\big]_{n,n}\Big], \tag{2.47}$$

where the expectation operator refers to independent random draws of the forecast ensemble $\mathbf{E}^\mathsf{f}$. In most geophysical systems, each state variable is attached to a specific position, called the **grid point**, in an underlying *physical* space, and the correlations decrease at a fast rate (*e.g.* exponentially) with the distance in the physical system. That is to say, if the n-th and m-th variables correspond to physically distant parts of the system,

$$\big[\mathbf{P}^\mathsf{f}_k\big]_{m,n} \approx 0, \tag{2.48}$$

while equation (2.47) becomes

$$\mathbb{E}\Big[\big[\bar{\mathbf{P}}^\mathsf{f}_k - \mathbf{P}^\mathsf{f}_k\big]^2_{m,n}\Big] \approx \frac{1}{N_\mathrm{e}-1}\Big[\big[\mathbf{P}^\mathsf{f}_k\big]_{m,m}\big[\mathbf{P}^\mathsf{f}_k\big]_{n,n}\Big]. \tag{2.49}$$

For a finite ensemble, if the variances of the n-th and m-th variables are non-zero, then $\bar{\mathbf{P}}^\mathsf{f}$ may exhibit a non-zero correlation between the n-th and m-th variables, a pattern which does not exist in $\mathbf{P}^\mathsf{f}$, as shown by equation (2.48).

This phenomenon is called *spurious correlation*. Fundamentally, it comes from the fact that the rank of $\bar{\mathbf{P}}^\mathsf{f}$ is limited by $N_\mathrm{e}-1$, the rank of the forecast perturbation matrix $\mathbf{X}^\mathsf{f}$. Therefore, $\bar{\mathbf{P}}^\mathsf{f}$ is a bad approximation of the potentially full-rank forecast error covariance matrix $\mathbf{P}^\mathsf{f}$ when the ensemble is small ($N_\mathrm{e} \ll N_\mathrm{x}$).

2.5.1.2 Negative bias of the analysis ensemble

By construction, the forecast ensemble $\mathbf{E}^\mathsf{f}$ has, on average, the correct sample covariance matrix. In other words,

$$\mathbb{E}\big[\bar{\mathbf{P}}^\mathsf{f}_k\big] = \mathbf{P}^\mathsf{f}_k, \tag{2.50}$$

where the expectation operator refers to independent random draws of $\mathbf{E}^\mathsf{f}$. As shown by Snyder (2014) in a simple example, the sample covariance matrix of the analysis ensemble $\mathbf{E}^\mathsf{a}$ obtained with the ETKF algorithm is (strictly) negatively biased:

$$\mathbb{E}\big[\operatorname{tr}\bar{\mathbf{P}}^\mathsf{a}_k\big] < \operatorname{tr}\mathbf{P}^\mathsf{a}_k. \tag{2.51}$$

Raanes et al. (2019a) explained that this result is not specific to the ETKF algorithm, but comes from the nonlinearity (concavity) of the map $\bar{\mathbf{P}}^\mathsf{f} \mapsto \bar{\mathbf{P}}^\mathsf{a}$.

2.5.2 Inflation

In order to counteract the strictly negative bias of the ensemble, as diagnosed by equation (2.51), a natural approach is to artificially inflate $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$. This can be performed in two different ways: either with multiplicative inflation as

$$\mathbf{E} \leftarrow \mathbf{E}\mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}} + \lambda\mathbf{E}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big), \tag{2.52}$$

or with additive inflation as

$$\mathbf{E} \leftarrow \mathbf{E} + \mathbf{Z}, \tag{2.53}$$

where the ensemble $\mathbf{E}$ is either $\mathbf{E}^{\mathsf{f}}$ or $\mathbf{E}^{\mathsf{a}}$, where $\mathbf{Z}$ is a an ensemble of random draws from a specified distribution, and where λ is the **multiplicative inflation factor**, a parameter to be determined. Raanes et al. (2019a) list the different motivations behind the use of inflation. Fundamentally, the objective is to compensate for different sources of errors and to improve the numerical stability of the algorithm.

In most numerical experiments, when the inflation is too weak the algorithm *diverges*[6] and when the inflation is too strong, the algorithm yields sub-optimal performances. The optimal inflation is very dependent on both the dynamical and observation system, and on the dedicated variant of the EnKF, and can even be inhomogeneous. In some situations, it is possible to try different implementations for the inflation and to select the one which yields the best performances. However, when the cost of performing a full DA experiment is high, it becomes impossible to tune the inflation. In this case, one can use adaptive inflation methods (*e.g.*, Raanes et al. 2019a), in which the optimal inflation is estimated on the fly.

2.5.3 Covariance localisation

In the EnKF, one of the key assumptions is that the best estimate of the forecast error covariance matrix $\mathbf{P}^{\mathsf{f}}$ is the forecast sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$. As remarked in section 2.5.1, this is probably a bad approximation, because $\mathbf{P}^{\mathsf{f}}$ can be full-rank while $\bar{\mathbf{P}}^{\mathsf{f}}$ has rank limited by $N_{\mathrm{e}} - 1$, and can include spurious correlations at long distance.

An empirical fix is to artificially mitigate the spurious correlations by regularising $\bar{\mathbf{P}}^{\mathsf{f}}$ as follows. The regularised forecast sample covariance matrix $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$ is defined as the element-wise multiplication[7] between $\bar{\mathbf{P}}^{\mathsf{f}}$ and $\boldsymbol{\rho} \in \mathbb{R}^{N_{\mathrm{x}}\times N_{\mathrm{x}}}$. The **localisation matrix $\boldsymbol{\rho}$** is a predefined short-range correlation matrix which is expected to represent the decay of correlations in the physical space. An important point is that the rank of $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$ is not any more limited by $N_{\mathrm{e}} - 1$. This approach is called covariance localisation (CL, Hamill et al. 2001).

Since $\bar{\mathbf{P}}^{\mathsf{f}}$ is positive semi-definite, the Schur product theorem (Horn and Johnson 2012) ensures that, if $\boldsymbol{\rho}$ is symmetric and positive semi-definite, then $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$ is symmetric and positive semi-definite as well. Most of the time, $\boldsymbol{\rho}$ is constructed as

$$\forall (m,n) \in (N_{\mathrm{x}}\!:\!1)^2, \quad [\boldsymbol{\rho}]_{m,n} = G\bigg(\frac{2d_{m,n}}{\ell}\bigg), \tag{2.54}$$

where $d_{m,n}$ is the physical distance between the m-th and n-th grid points, ℓ is the localisation radius, a parameter to be determined, and G is the fifth-order piecewise rational Gaspari–Cohn

[6] By divergence, it is meant that the difference between the truth $\mathbf{x}^{\mathrm{t}}$ and the sample mean $\bar{\mathbf{x}}$ is much larger than could be expected from the sample covariance matrix $\bar{\mathbf{P}}$.

[7] Also called Schur or Hadamard product.

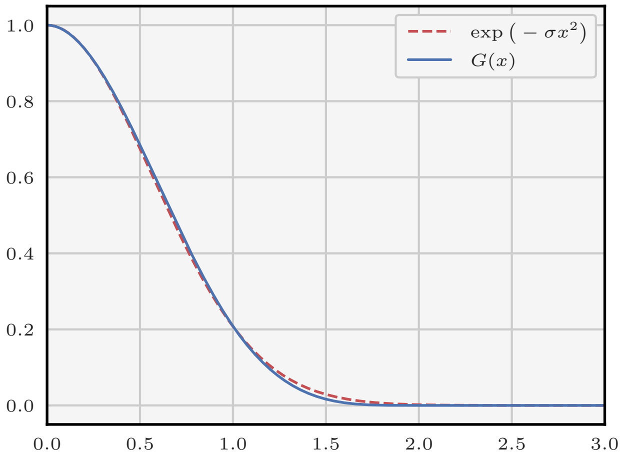

Figure 2.1: Fifth-order piecewise rational GC function, defined by equation (2.55), in blue. The closest Gaussian density, $\exp\left(-\sigma x^2\right)$ with $\sigma \approx 1.57$, is plotted in red for comparison.

(GC) function (Gaspari and Cohn 1999), defined by

$$G : \begin{cases} \mathbb{R}_+ & \to \mathbb{R}_+, \\ x \in [0,1] & \mapsto 1 - \frac{5}{3}x^2 + \frac{5}{8}x^3 + \frac{1}{2}x^4 - \frac{1}{4}x^5, \\ x \in [1,2] & \mapsto 4 - 5x + \frac{5}{3}x^2 + \frac{5}{8}x^3 - \frac{1}{2}x^4 + \frac{1}{12}x^5 - \frac{2}{3x}, \\ x \geq 2 & \mapsto 0, \end{cases} \tag{2.55}$$

and illustrated in figure 2.1. Visually, the GC function is similar to a Gaussian density, but it has a compact support.

When using CL, the EnKF equations derived in ensemble space are not valid any more. This means that one has to use the EnKF equations derived in state space, which *a priori* makes CL inapplicable to high-dimensional systems. This is further discussed in part III, which is dedicated to the implementation of CL in the EnKF.

Remark 9. The factor 2 in the right-hand side of equation (2.54) ensures that the distance at which correlations fall to zero is indeed ℓ. It comes from the fact that the support of the GC function G is $[0,2]$ instead of $[0,1]$.

2.5.4 Domain localisation

Suppose that each component y of the observation vector $\mathbf{y}$, simply called **observation**, is attached to a specific position in the physical space, called the **site** of the observation.

2.5.4.1 Principle of domain localisation in EnKFs

Instead of mitigating the spurious correlations, another route to localisation is to limit the influence of each observation in the analysis step to a local neighbourhood of its site. This strategy implies that the analysis step must be local (*i.e.* each state variable is independently updated), and the observation error covariance matrix $\mathbf{R}$ must be amended accordingly. This approach is called domain localisation (DL, Houtekamer and Mitchell 2001; Ott et al. 2004).

A possible implementation of the n-th local analysis is to define the tapered observation error covariance matrix $\mathbf{R}_n$ using an element-wise multiplication between the **observation precision** matrix $\mathbf{R}^{-1}$ and $\boldsymbol{\rho}_n \in \mathbb{R}^{N_\mathrm{y}\times N_\mathrm{y}}$, a predefined short-range correlation matrix:

$$\mathbf{R}_n^{-1} \triangleq \boldsymbol{\rho}_n \circ \mathbf{R}^{-1}. \tag{2.56}$$

The localisation matrix $\boldsymbol{\rho}_n$ is expected to represent the decay of correlations relative to the n-th grid point, and can be constructed as

$$\forall (p,q) \in (N_\mathrm{y}:1)^2, \quad [\boldsymbol{\rho}_n]_{p,q} = \sqrt{G\left(\frac{2d_{p,n}}{\ell}\right)}\sqrt{G\left(\frac{2d_{q,n}}{\ell}\right)}, \tag{2.57}$$

where $d_{p,n}$ is the physical distance between the p-th site and the n-th grid point, $d_{q,n}$ is the physical distance between the q-th site and the n-th grid point, and ℓ is the localisation radius. As a consequence, the n-th **local domain**, defined as the set of all observations which contribute to the n-th local analysis, is only composed of all observations whose site is located within distance ℓ to the n-th grid point. Finally, from the resulting analysis ensemble $\mathbf{E}^\mathsf{a}$, only the n-row (which corresponds to the n-th state variable) is kept. The main idea behind DL is that the forecast sample covariance matrix $\bar{\mathbf{P}}^\mathsf{f}$ is locally full-rank, meaning that the ensemble can accommodate the information content in each local domain (Carrassi et al. 2018).

The Schur product theorem once again ensures that the tapered observation error covariance matrices $\mathbf{R}_{N_\mathrm{x}:1}$ are positive semi-definite, which is sufficient for the equations of the EnKF derived in ensemble space to remain valid. This is the major advantage of DL over CL: each local analysis is implemented exactly as the original (global) analysis but using a different $\mathbf{R}$. For example, algorithm 2.4 describes a full assimilation cycle for the local ensemble transform Kalman filter (LETKF) algorithm, a variant of the ETKF algorithm in which DL is implemented (Hunt et al. 2007). The similarities between algorithms 2.3 and 2.4 are clearly apparent.

The global part of the algorithm is unchanged, and still has algorithmic complexity $\mathcal{O}(T_\mathbf{H} N_\mathrm{e})$. On the other hand, the algorithmic complexity of each local analysis has been reduced from $\mathcal{O}\big(N_\mathrm{y}^2 N_\mathrm{e} + N_\mathrm{x} N_\mathrm{e}^2\big)$ to $\mathcal{O}\big(N_\mathrm{y}^2 N_\mathrm{e}\big)$. If the $\boldsymbol{\rho}_{N_\mathrm{x}:1}$ are sparse, as can be expected from equation (2.57), only a subset of observations contribute to a specific local analysis. This information can be exploited to further reduce the cost of each local analysis to[8] $\mathcal{O}\big((N_\mathrm{y}^\ell)^2 N_\mathrm{e}\big)$, where N_y^ℓ is the maximum number of observations in each local domain, given by

$$N_\mathrm{y}^\ell = \max_{n\in N_\mathrm{x}:1} \mathrm{Card}\Big\{ q \in (N_\mathrm{y}:1) \setminus \exists p \in (N_\mathrm{y}:1) \setminus \big[\boldsymbol{\rho}_n\big]_{p,q} \neq 0 \Big\}. \tag{2.58}$$

[8] For simplicity, it is assumed here that $N_\mathrm{y}^\ell \geq N_\mathrm{e}$.

Algorithm 2.4: Full assimilation cycle for the LETKF algorithm in the context of the GL system. In steps 14 and 15, $\bar{x}^{\mathsf{f}}_n$ and $\bar{x}^{\mathsf{a}}_n$ designate the n-th element of the vectors $\bar{\mathbf{x}}^{\mathsf{f}}$ and $\bar{\mathbf{x}}^{\mathsf{a}}$, and $\mathbf{X}^{\mathsf{f}}_n$ and $\mathbf{X}^{\mathsf{a}}_n$ designate the n-th row of the matrices $\mathbf{X}^{\mathsf{f}}$ and $\mathbf{X}^{\mathsf{a}}$.

```
   Input: $\mathbf{E}^{\mathsf{a}}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
   Parameters: $\mathbf{M}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$, $\boldsymbol{\rho}_{N_\mathrm{x}:1}$
 1 Forecast
 2     $\bar{\mathbf{x}} \leftarrow \mathbf{E}^{\mathsf{a}}\mathbf{1}/N_\mathrm{e}$
 3     $\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{a}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_\mathrm{e}\big)/\sqrt{N_\mathrm{e} - 1}$
 4     $\mathbf{Z} \leftarrow (\mathbf{M}\mathbf{X})^{+}$
 5     $\mathbf{T}_\mathrm{e} \leftarrow \mathbf{I} + \mathbf{Z}\mathbf{Q}\mathbf{Z}$
 6     $\bar{\mathbf{x}}^{\mathsf{f}} \leftarrow \mathbf{M}\bar{\mathbf{x}}$
 7     $\mathbf{X}^{\mathsf{f}} \leftarrow \mathbf{M}\mathbf{X}\mathbf{T}_\mathrm{e}^{1/2}$
 8 Analysis
 9     $\mathbf{Y} \leftarrow \mathbf{H}\mathbf{X}^{\mathsf{f}}$
10     $\mathbf{d} \leftarrow \mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^{\mathsf{f}}$
11     for $n = 1$ to $N_\mathrm{x}$ do
12         $\mathbf{R}_n^{-1} \leftarrow \boldsymbol{\rho}_n \circ \mathbf{R}^{-1}$
13         $\mathbf{T}_\mathrm{e} \leftarrow \mathbf{I} + \mathbf{Y}^{\mathsf{T}}\mathbf{R}_n^{-1}\mathbf{Y}$
14         $\mathbf{w} \leftarrow \mathbf{T}_\mathrm{e}^{-1}\mathbf{Y}^{\mathsf{T}}\mathbf{R}_n^{-1}\mathbf{d}$
15         $\bar{x}^{\mathsf{a}}_n \leftarrow \bar{x}^{\mathsf{f}}_n + \mathbf{X}^{\mathsf{f}}_n\mathbf{w}$
16         $\mathbf{X}^{\mathsf{a}}_n \leftarrow \mathbf{X}^{\mathsf{f}}_n\big(\mathbf{T}_\mathrm{e}\big)^{-1/2}$
17     end
18     $\mathbf{E}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{a}}\mathbf{1}^{\mathsf{T}} + \sqrt{N_\mathrm{e} - 1}\mathbf{X}^{\mathsf{a}}$
   Output: $\mathbf{E}^{\mathsf{a}}\ [t_{k+1}]$
```

If the $\boldsymbol{\rho}_{N_\mathrm{x}:1}$ are constructed with equation (2.57), then N^{ℓ}_{y} is given by

$$N^{\ell}_{\mathrm{y}} = \max_{n\in(N_\mathrm{x}:1)} \mathrm{Card}\big\{q \in (N_\mathrm{y}:1) \setminus d_{q,n} \leq \ell\big\}. \tag{2.59}$$

Taken into account the number of local analyses, the total algorithmic complexity of the local part of the LETKF algorithm is $\mathcal{O}\big(N_\mathrm{x}(N^{\ell}_{\mathrm{y}})^2 N_\mathrm{e}\big)$. However, the local analyses are embarrassingly parallel. Therefore, this algorithmic complexity can be reduced by a factor N_t, the number of threads running in parallel.

2.5.4.2 Domain localisation and imbalance

The state vector $\boldsymbol{x}$ is a trajectory of the dynamical system. As a consequence it is expected to be on the attractor of the dynamical system. In complex geophysical models, this is explained by a certain regularity in the fields and by elaborate balances between variables. In this thesis, **imbalance** is defined as a measure of the distance to the attractor of the dynamical system. Imbalance is a major preoccupation when using DL, because on the one hand, ensemble members are supposed to represent possible realisations of the (balanced) state vector $\boldsymbol{x}$. On the other hand, with DL the (global) analysis ensemble is obtained by assembling potentially very different local analysis ensembles, which can lead to imbalance (Kepert 2009; Greybush et al. 2011).

Suppose that the localisation matrices $\boldsymbol{\rho}_{N_\mathrm{x}:1}$ are controlled by a localisation radius ℓ, as in equation (2.57). If the localisation radius ℓ is large, then the tapered observation error covariance matrices $\mathbf{R}_{N_\mathrm{x}:1}$ vary smoothly from one grid point to another, thanks to the continuity of the GC function. As a consequence, the local analysis ensembles $\mathbf{E}^{\mathsf{a}}_{N_\mathrm{x}:1}$ varies smoothly from one grid point to another, and this should limit imbalance. However if the localisation radius ℓ is too large, then the added value of localisation is null. Now if the localisation radius ℓ is small, then the information contained in the local domains is small and can be handled by small ensembles. On the other hand if the localisation radius ℓ is too small, then the local analysis ensembles $\mathbf{E}^{\mathsf{a}}_{N_\mathrm{x}:1}$ are very different from one grid point to the other, and the resulting imbalance can be problematic when applying the dynamical model $\mathbf{M}$ in the next forecast step.

When the dynamical system is chaotic, localisation has been shown to be mandatory to avoid the divergence when the ensemble size N_e is smaller than or equal to the number of unstable and neutral modes of the dynamics (Bocquet and Carrassi 2017). Again, the optimal localisation is very dependent on both the dynamical and observation systems, and on the dedicated variant of the EnKF, and can even be inhomogeneous. In some situations, it is possible to try different implementations of the localisation, and to select the one which yields the best performances. However, when the cost of performing a full DA experiment is high, it becomes impossible to tune the localisation. In this case, one can use preliminary statistical studies to determine the optimal localisation for a given DA system in a given regime (Anderson and Lei 2013). Alternatively, one can use adaptive localisation methods, in which the optimal localisation is estimated on the fly (Ménétrier et al. 2015a,b).

2.5.4.3 Connection between covariance and domain localisation

Both CL and DL are based on the same property, the decay of correlations, but their approach to localisation is different and lead to distinct implementations. On the one hand, DL relies on a collection of local analyses, which are by construction embarrassingly parallel. On the the other hand, CL relies on a single global analysis with a localised forecast error covariance matrix $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$, but for which there is no obvious parallelisation of the analysis step. In a more general perspective, DL is applied in state space, while CL is applied in observation space. In particular, an important prerequisite of DL is that each observation y must have a well-defined site. This means that, with DL, non-local observations cannot be assimilated without *ad hoc* approximations.

Although the connection between CL and DL is not obvious, they have been shown to yield equivalent results in the limit of *weak* data assimilation, when the forecast density π^{f} is highly informative (Sakov and Bertino 2011).

3 The particle filter

Contents

In the statistical literature, MC methods are very attractive, because they enable the computation of very complex integrals using a random draws. The convergence of the method is ensured by the law of large numbers, and performance bounds can be estimated using the central limit theorem.

In stochastic EnKF algorithms, MC methods are used in the forecast step to propagate the ensemble, and in the analysis step to obtain an ensemble of perturbed observations. Going one step further, one can try solving the filtering estimation problem using only MC methods. This leads to the famous sequential importance sampling (SIS) algorithm (see, *e.g.*, Doucet et al. 2001, and references therein). The particle filter (PF) is a class of filtering DA algorithms based on the SIS algorithm, the most famous example being the sampling importance resampling (SIR) algorithm (Rubin 1987; Gordon et al. 1993), also known as Bootstrap filter. Although the PF shares common ideas with the EnKF – both are ensemble-based filtering DA methods – three fundamental differences should be reported. First, the PF does not rely on linear and Gaussian hypotheses, and it solves the generic filtering estimation problem, problem 1.1, in a sense to be defined in section 3.5. Second, in general the PF does not involve linear algebra, and therefore it leads to simpler and faster algorithms than the EnKF. And third, for a successful application, the PF requires an ensemble size which scales exponentially with the dimension of the DA system. This phenomenon is known in the literature as the *curse of dimensionality* (see, *e.g.*, Snyder et al. 2008), and it means that, as is, the PF is inapplicable to high-dimensional DA systems. This is why, in a sense, the PF can be seen as a brute force approach to DA.

This chapter gives an introduction to the PF, and is inspired from the following references: Arulampalam et al. (2002), van Leeuwen (2009), Doucet and Johansen (2011) and van Leeuwen et al. (2019). Section 3.1 describes general aspect of MC methods, and introduces the SIS algorithm. Section 3.2 presents the core elements of the PF. Sections 3.3 and 3.4 discuss the resampling step and the proposal density of the PF. Section 3.5 concludes this chapter with an overview of the main convergence results.

3.1 Monte Carlo methods for data assimilation

We start this chapter with a general presentation of MC methods and how they can be used to solve the generic filtering estimation problem, problem 1.1.

3.1.1 Unbiased Monte Carlo sampling

For simplicity, the time evolution of the generic DA system is temporarily put aside. Let $\mathcal{F} : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$ be a π^a-integrable[1] test function and suppose that the goal is to estimate the expectation F of $\mathcal{F}$ over π^a, defined as[2]

$$F \triangleq \mathbb{E}\big[\mathcal{F}(\boldsymbol{x}|\boldsymbol{y})\big] = \int \mathcal{F}(\mathbf{x})\,\pi^\mathsf{a}(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x}. \tag{3.1}$$

Suppose that it is possible to produce an iid sample $\mathsf{E} = \big\{\mathbf{x}_i \in \mathbb{R}^{N_\mathrm{x}}, i \in (N_\mathrm{e}\!:\!1)\big\}$ from the analysis distribution ν^a. Then, F can be approximated using the unbiased MC estimate $\widehat{F}$, defined as

$$\widehat{F} \triangleq \frac{1}{N_\mathrm{e}}\sum_{i=1}^{N_\mathrm{e}} \mathcal{F}\big(\mathbf{x}_i\big). \tag{3.2}$$

From the strong law of large numbers we deduce that $\widehat{F}$ almost surely converges towards F in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$. Moreover, if $\mathcal{F}^2$ is π^a-integrable as well, the variance of $\mathcal{F}$ can be defined as

$$\sigma^2 \triangleq \mathbb{V}\big[\mathcal{F}(\boldsymbol{x}|\boldsymbol{y})\big] = \int \mathcal{F}^2(\mathbf{x})\,\pi^\mathsf{a}(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x} - F^2. \tag{3.3}$$

[1] In this chapter, integrability conditions are necessary to ensure that all integrals are well-defined.

[2] In this chapter, unless specified otherwise, integrations are performed over the whole state space $\mathbb{R}^{N_\mathrm{x}}$.

In the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the central limit theorem ensures that the rescaled error $\sqrt{N_\mathrm{e}}\big(\widehat{F} - F\big)$, converges in distribution towards $\mathcal{N}\big[0, \sigma^2\big]$.[3]

By definition, the variance of $\widehat{F}$ is given by

$$\mathbb{V}\Big[\widehat{F}\Big] = \mathbb{V}\Big[\widehat{F} - F\Big] = \mathbb{E}\Big[\big(\widehat{F} - F\big)^2\Big] - \mathbb{E}\Big[\widehat{F} - F\Big]^2, \tag{3.4}$$

where the expectation and variance operators refers to independent random draws of the ensemble E. This means that the mean-squared error of the unbiased MC estimate $\widehat{F}$ is the sum of its variance and its squared bias. Using the weak law of large numbers, we deduce that the bias term is null. Finally using equation (3.2), the variance term is given by

$$\mathbb{V}\Big[\widehat{F}\Big] = \frac{\sigma^2}{N_\mathrm{e}}. \tag{3.5}$$

The convergence results for this unbiased MC method are formalised by theorems 3.1 and 3.2. However in practice, the analysis distribution ν^a is complex, and only known up to a proportionality constant. Therefore, we cannot use this unbiased MC method.

Theorem 3.1 (Strong law of large numbers)**.** *For any π^a-integrable test function $\mathcal{F} : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$, the unbiased MC estimate $\widehat{F}$ is an unbiased estimate of the expectation F, and almost surely converges towards the expectation F in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$.*

Theorem 3.2 (Central limit theorem)**.** *Furthermore, if $\mathcal{F}^2$ is π^a-integrable, then the rescaled error $\sqrt{N_\mathrm{e}}\big(\widehat{F} - F\big)$ converges in distribution towards $\mathcal{N}\big[0, \sigma^2\big]$ in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, where the asymptotic variance σ^2 is the variance of the test function $\mathcal{F}$, given by equation* (3.3)*.*

3.1.2 Importance sampling

We introduce the **proposal** vector $\boldsymbol{v}$, a random vector with N_x elements whose distribution $\nu^\mathsf{q} \triangleq \nu[\boldsymbol{v}]$ is easy to sample from, and whose pdf $\pi^\mathsf{q} \triangleq \pi[\boldsymbol{v}]$ has a larger support than π^a. For example, the proposal distribution ν^q can be a multivariate Gaussian distribution. Using the (perfect) **importance weight** function $\widetilde{w}$, defined as

$$\widetilde{w}(\mathbf{x}) \triangleq \frac{\pi^\mathsf{a}(\mathbf{x}|\mathbf{y})}{\pi^\mathsf{q}(\mathbf{x})}, \tag{3.6}$$

it is possible to compute F as

$$F = \int \mathcal{F}(\mathbf{x})\,\widetilde{w}(\mathbf{x})\,\pi^\mathsf{q}(\mathbf{x})\,\mathrm{d}\mathbf{x} = \mathbb{E}\big[(\mathcal{F}\widetilde{w})(\boldsymbol{v})\big]. \tag{3.7}$$

[3] More formally, it is meant here that the rescaled error $\sqrt{N_\mathrm{e}}\big(\widehat{F} - F\big)$, seen as a random variable, converges in distribution towards the random variable distributed according to $\mathcal{N}\big[0, \sigma^2\big]$.

This means that the unbiased MC method described in the previous subsection could be applied to obtain the unbiased importance sampling (IS) estimate $\widetilde{F}$, defined as

$$\widetilde{F} = \frac{1}{N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} \mathcal{F}(\mathbf{x}_i)\, \widetilde{w}(\mathbf{x}_i), \tag{3.8}$$

where the ensemble $\mathsf{E} = \big\{\mathbf{x}_i \in \mathbb{R}^{N_\mathrm{x}}, i \in (N_\mathrm{e} : 1)\big\}$ is now an iid sample from the proposal distribution ν^q.

The unbiased IS estimate $\widetilde{F}$ satisfies the exact same properties as the unbiased MC estimate $\widehat{F}$. However, using Bayes' theorem, the analysis density is equal to

$$\pi^\mathsf{a}(\mathbf{x}|\mathbf{y}) = \frac{\pi^\mathsf{o}(\mathbf{y}|\mathbf{x})\, \pi^\mathsf{b}(\mathbf{x})}{\pi[\boldsymbol{y}](\mathbf{y})}, \tag{3.9}$$

in which the normalisation constant $\pi[\boldsymbol{y}](\mathbf{y})$ is often unknown. Therefore, it seems more realistic to define the (unnormalised) importance weight function w as

$$w(\mathbf{x}) \triangleq \frac{\pi^\mathsf{o}(\mathbf{y}|\mathbf{x})\, \pi^\mathsf{b}(\mathbf{x})}{\pi^\mathsf{q}(\mathbf{x})}. \tag{3.10}$$

The normalisation constant $\pi[\boldsymbol{y}](\mathbf{y})$ can then be computed as

$$\pi[\boldsymbol{y}](\mathbf{y}) = \int \pi^\mathsf{o}(\mathbf{y}|\mathbf{x})\, \pi^\mathsf{b}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int w(\mathbf{x})\, \pi^\mathsf{q}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \mathbb{E}\big[w(\boldsymbol{v})\big]. \tag{3.11}$$

This means that equation (3.7) becomes

$$F = \frac{\mathbb{E}\big[(\mathcal{F}w)(\boldsymbol{v})\big]}{\mathbb{E}\big[w(\boldsymbol{v})\big]}. \tag{3.12}$$

Therefore, F can be approximated using the IS estimate $\bar{F}$, defined as

$$\bar{F} \triangleq \frac{\sum_{i=1}^{N_\mathrm{e}} \mathcal{F}(\mathbf{x}_i)\, w(\mathbf{x}_i)}{\sum_{i=1}^{N_\mathrm{e}} w(\mathbf{x}_i)}, \tag{3.13}$$

in which the ensemble $\mathsf{E} = \{\mathbf{x}_i, i \in (N_\mathrm{e} : 1)\}$ is an iid sample from the proposal distribution ν^q. In this context, ensemble members are usually called **particles** and $w(\mathbf{x}_i)$ is the (unnormalised) importance weight of the i-th particle.

Unlike $\widehat{F}$ and $\widetilde{F}$, the IS estimate $\bar{F}$ is biased. The bias here comes from the fact that the normalisation constant needs to be estimated. If $\widetilde{w}$ and $\mathcal{F}\widetilde{w}$ are π^a-integrable, then in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the asymptotic rescaled bias b_∞ is given by

$$b_\infty \triangleq \lim_{N_\mathrm{e}\to\infty} N_\mathrm{e}\big(\mathbb{E}\big[\bar{F}\big] - F\big) = -\mathbb{E}\Big[\widetilde{w}(\boldsymbol{x}|\boldsymbol{y})\big[\mathcal{F}(\boldsymbol{x}|\boldsymbol{y}) - F\big]\Big] \tag{3.14a}$$

$$= -\int \widetilde{w}(\mathbf{x})\big[\mathcal{F}(\mathbf{x}) - F\big]\pi^\mathsf{a}(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x} \tag{3.14b}$$

where $\mathbb{E}\big[\bar{F}\big]$ refers to independent random draws of the ensemble E (Doucet and Johansen 2011). Nevertheless, the strong law of large numbers ensures that, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, $\bar{F}$ almost surely converges towards F. Furthermore, if both $\widetilde{w}$ and $\mathcal{F}^2\widetilde{w}$ are π^a-integrable, then theorem 2 of Geweke (1989) ensures that, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the rescaled error $\sqrt{N_\mathrm{e}}\big(\bar{F} - F\big)$ converges in distribution towards $\mathcal{N}\big[0, \sigma^2\big]$, where the asymptotic variance σ^2 is given by

$$\sigma^2 \triangleq \mathbb{E}\Big[\widetilde{w}(\boldsymbol{x}|\boldsymbol{y})\big[\mathcal{F}(\boldsymbol{x}|\boldsymbol{y}) - F\big]^2\Big] = \int \widetilde{w}(\mathbf{x})\big[\mathcal{F}(\mathbf{x}) - F\big]^2\pi^\mathsf{a}(\mathbf{x}|\mathbf{y})\,\mathrm{d}\mathbf{x}. \tag{3.15}$$

Both the variance and the bias of the IS estimate $\bar{F}$ are inversely proportional to the ensemble size N_e. This means that the mean-squared error of $\bar{F}$ is asymptotically dominated by its variance.

The major advantage of IS method is that, contrary to the unbiased MC method, it can be directly applied to a realistic DA system. The convergence results for this IS method are formalised by theorems 3.3 and 3.4.

Theorem 3.3 (Strong law of large numbers for IS)**.** *For any π^a-integrable test function $\mathcal{F} : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$, the IS estimate $\bar{F}$ is a biased estimate of the expectation F, and almost surely converges towards the expectation F in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$.*

Theorem 3.4. *Furthermore, if both w and $\mathcal{F}^2 w$ are π^a-integrable, then the rescaled error $\sqrt{N_\mathrm{e}}\big(\bar{F} - F\big)$ converges in distribution towards $\mathcal{N}\big[0, \sigma^2\big]$ in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, where the asymptotic variance σ^2 is given by equation* (3.15)*.*

The relative efficiency of the IS method compared to the unbiased MC method can be measured by the ratio between the variances of $\widehat{F}$ and of $\bar{F}$. In general, this ratio depends on the test function $\mathcal{F}$. However, by keeping only the first two statistical moments, it is possible to show that

$$\frac{\mathbb{V}\big[\bar{F}\big]}{\mathbb{V}\big[\widehat{F}\big]} \approx 1 + \mathbb{V}\big[w(\boldsymbol{v})\big], \tag{3.16}$$

where the variance operators in the left-hand-side refer to independent random draws of the ensemble E from the proposal distribution ν^q for $\bar{F}$ and from the analysis distribution ν^a for $\widehat{F}$ (Kong et al. 1994). This means that, in order to obtain accurate estimates with the IS method, it is a good strategy to use a proposal distribution ν^q which minimises the variance of the importance weight function.

Remark 10. In the statistical literature, the IS estimate $\bar{F}$ is said to be *consistent*. However, in order to avoid any confusion with the consistency terminology introduced in subsection 2.2.2, this terminology is not used here.

In order to illustrate the statistical properties of IS, consider the following one-dimensional example, in which the background- and observation densities π^{b} and π^{o} are given by

$$\pi^{\mathsf{b}}(x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\left(x - x^{\mathsf{b}}\right)^2}{2}\right], \tag{3.17a}$$

$$\pi^{\mathsf{o}}(y|x) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{\left(x - y\right)^2}{2}\right], \tag{3.17b}$$

with $x^{\mathsf{b}} = -1$ and $y = 1$. The analysis density π^{a}, obtained using Bayes' theorem, is given by

$$\pi^{\mathsf{a}}(x|y) = \frac{1}{\sqrt{\pi}} \exp\left[-\left(x - x^{\mathsf{a}}\right)^2\right], \tag{3.17c}$$

with $x^{\mathsf{a}} = 0$. Suppose that we are using IS with the background as proposal (in other words $\pi^{\mathsf{q}} = \pi^{\mathsf{b}}$), and that the test function $\mathcal{F}$ is the identity $x \mapsto x$, whose expected value is $F = x^{\mathsf{a}} = 0$.

In this case, the asymptotic bias b_∞ and the asymptotic variance σ^2, given by equations (3.14b) and (3.15), are equal to

$$-b_\infty = \frac{2\sqrt{3}}{9} \exp\frac{2}{3} \approx 0.75, \tag{3.18}$$

$$\sigma^2 = \frac{8\sqrt{3}}{27} \exp\frac{2}{3} \approx 1.00. \tag{3.19}$$

Figure 3.1 illustrates the pdfs of this problem. Figure 3.2 shows the evolution of the rescaled bias $N_\mathrm{e}\big(\mathbb{E}\big[\bar{F}\big] - F\big)$ as a function of the ensemble size N_e. Finally, figure 3.3 shows the empirical density of the rescaled error $\sqrt{N_\mathrm{e}}\big(\bar{F} - F\big)$ for several values of the ensemble size N_e.

3.1.3 Sequential importance sampling

Without loss of generality, it is possible to apply the IS method to the generic DA system with time evolution. In this case, at time t_k, the target distribution is the full conditional distribution $\nu[\boldsymbol{x}_{k:}|\boldsymbol{y}_{k:}]$, and each particle $\mathbf{x}_i$ represents a possible trajectory of the system between t_0 and t_k:[4]

$$\mathbf{x}_i = \big(\mathbf{x}_i(l), l \in (k:)\big). \tag{3.20}$$

Now suppose that, in a similar way as the joint density $\pi[\boldsymbol{x}_{k:}, \boldsymbol{y}_{k:}]$ with equation (1.1), the proposal density $\pi[\boldsymbol{v}_{k:}]$ can be factored as

$$\pi[\boldsymbol{v}_{k:}] = \pi[\boldsymbol{v}_0] \prod_{l=1}^{k} \pi[\boldsymbol{v}_l|\boldsymbol{v}_{l-1}]. \tag{3.21}$$

[4] In order to avoid double subscripts in this chapter, a functional notation is used instead of a subscript notation for the time indices if necessary.

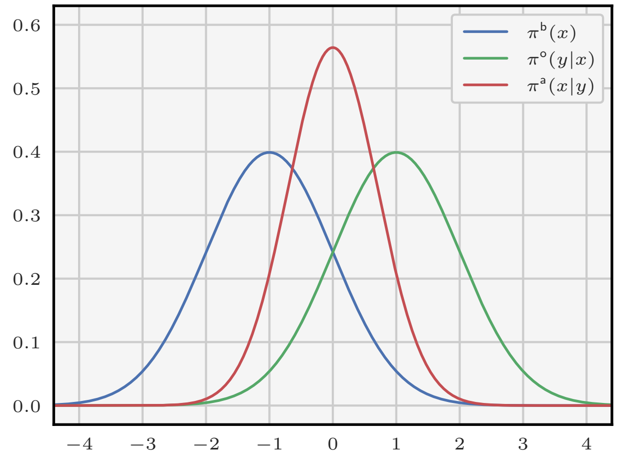

Figure 3.1: Illustration of the background density π^{b} (in blue), of the observation density π^{o} (in green), and of the analysis density π^{a} (in red) for the one-dimensional example considered in subsection 3.1.2.

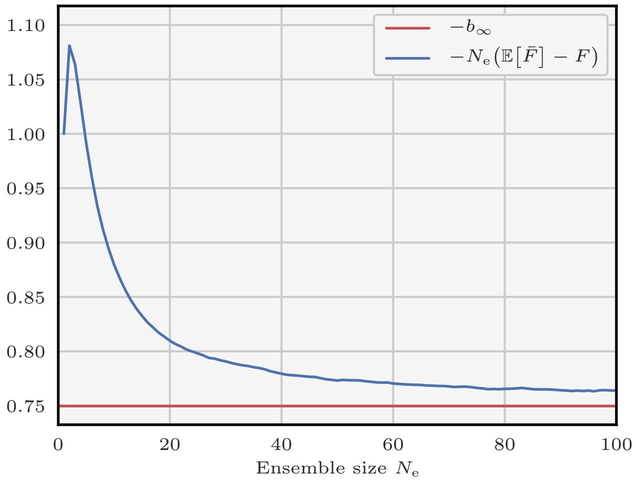

Figure 3.2: Evolution of the rescaled bias $-N_{\mathrm{e}}\big(\mathbb{E}\big[\bar{F}\big] - F\big)$ as a function of the ensemble size N_{e} (in blue) for the one-dimensional example considered in subsection 3.1.2. The expectation operator is approximated by 10^7 independent random draws of the ensemble E according to the background distribution ν^{b}. The asymptotic bias $-b_{\infty}$ is shown with an horizontal red line.

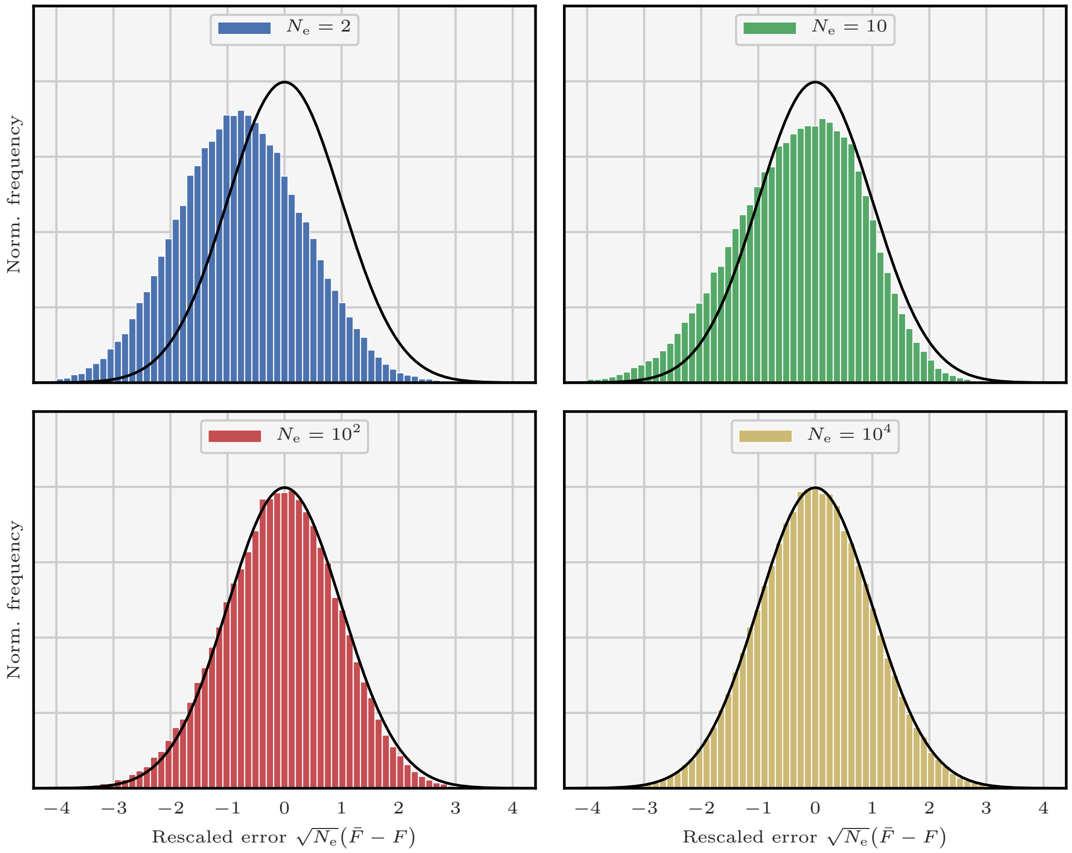

Figure 3.3: Empirical density of the rescaled error $\sqrt{N_e}(\bar{F} - F)$ for $N_e = 2$ (top-left panel, in blue), $N_e = 10$ (top-right panel, in green), $N_e = 10^2$ (bottom-left panel, in red), and $N_e = 10^4$ (bottom-right panel, in yellow) in the one-dimensional example considered in subsection 3.1.2. The histograms are computed using 10^5 independent random draws of the ensemble E from the background distribution ν^{b}. The asymptotic density in the limit of an infinite ensemble, $N_e \to \infty$, is plotted in black and can hardly be distinguished from the case $N_e = 10^4$.

Then, the importance weight function w is given by

$$w_k(\mathbf{x}_{k:}) = \frac{\pi[\boldsymbol{x}_{k:}, \boldsymbol{y}_{k:}](\mathbf{x}_{k:}, \mathbf{y}_{k:})}{\pi[\boldsymbol{v}_{k:}](\mathbf{x}_{k:})}. \tag{3.22}$$

For convenience, we introduce the following notation for the proposal distribution:

$$\nu^{\mathsf{qb}} \triangleq \nu[\boldsymbol{v}_0], \tag{3.23a}$$

$$\nu^{\mathsf{q}}_k[\mathbf{x}_k] \triangleq \nu[\boldsymbol{v}_{k+1}|\boldsymbol{v}_k = \mathbf{x}_k], \tag{3.23b}$$

for the proposal density:

$$\pi^{\mathsf{qb}} \triangleq \pi[\boldsymbol{v}_0], \tag{3.23c}$$

$$\pi^{\mathsf{q}}_k \triangleq \pi[\boldsymbol{v}_{k+1}|\boldsymbol{v}_k], \tag{3.23d}$$

and for the **incremental** importance weight function:

$$w^{\mathsf{i}}_0(\mathbf{x}_0) \triangleq \frac{\pi^{\mathsf{o}}_0(\mathbf{y}_0|\mathbf{x}_0)\,\pi^{\mathsf{b}}(\mathbf{x}_0)}{\pi^{\mathsf{qb}}(\mathbf{x}_0)}, \tag{3.23e}$$

$$w^{\mathsf{i}}_{k+1}(\mathbf{x}_{k+1}|\mathbf{x}_k) \triangleq \frac{\pi^{\mathsf{o}}_{k+1}(\mathbf{y}_{k+1}|\mathbf{x}_{k+1})\,\pi^{\mathsf{m}}_k(\mathbf{x}_{k+1}|\mathbf{x}_k)}{\pi^{\mathsf{q}}_k(\mathbf{x}_{k+1}|\mathbf{x}_k)}. \tag{3.23f}$$

Using this notation, the importance weight function can be computed through the recursion

$$w_{k+1}(\mathbf{x}_{k+1:}) = w_k(\mathbf{x}_{k:})\, w^{\mathsf{i}}_{k+1}(\mathbf{x}_{k+1}|\mathbf{x}_k). \tag{3.24}$$

Therefore, in order to compute the particles and their importance weights, one can use a recursive algorithm, such as algorithm 3.1, known in the statistical literature as the SIS algorithm (see, *e.g.*, Doucet et al. 2001).

Let $\mathsf{E} = \big\{(w_i, \mathbf{x}_i), i \in (N_\mathrm{e}:1)\big\}$ be the weighted ensemble at time t_k resulting from the SIS algorithm, with

$$\forall i \in (N_\mathrm{e}:1), \quad w_i = \big(w_i(l), l \in (k:)\big), \tag{3.25a}$$

$$\forall i \in (N_\mathrm{e}:1), \quad \mathbf{x}_i = \big(\mathbf{x}_i(l), l \in (k:)\big). \tag{3.25b}$$

In this thesis, the focus is on the analysis density π^{a} (filtering density, as opposed to the smoothing density). This is why for any $l \in (k:)$ and any π^{a}_l-integrable test function $\mathcal{F}_l : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$, we define the expectation F_l and the IS estimate $\bar{F}_l$ as

$$F_l \triangleq \mathbb{E}\big[\mathcal{F}_l(\boldsymbol{x}_l|\boldsymbol{y}_{l:})\big], \tag{3.26}$$

$$\bar{F}_l \triangleq \sum_{i=1}^{N_\mathrm{e}} \mathcal{F}_l\big(\mathbf{x}_i(l)\big)\,\bar{w}_i(l), \tag{3.27}$$

where the normalised importance weights $\bar{w}_{N_\mathrm{e}:1}$ are defined by

$$\forall i \in (N_\mathrm{e}:1), \quad \bar{w}_i(l) \triangleq \frac{w_i(l)}{\sum\limits_{j=1}^{N_\mathrm{e}} w_j(l)}. \tag{3.28}$$

Algorithm 3.1: SIS algorithm for the generic DA system.

```
Input: $\mathbf{y}\ [t_0 \rightarrow t_k]$
Parameters: $w^{\mathsf{i}}_{k:}$, $\nu^{\mathsf{qb}}$, $\nu^{\mathsf{q}}_{k-1:}$
for $i = 1$ to $N_{\mathrm{e}}$ do
    $\mathbf{x}_i(0) \sim \nu^{\mathsf{qb}}$
    $w_i(0) \leftarrow w^{\mathsf{i}}_0\big(\mathbf{x}_i(0)\big)$
    for $l = 0$ to $k-1$ do
        $\mathbf{x}_i(l+1) \sim \nu^{\mathsf{q}}_l\big[\mathbf{x}_i(l)\big]$
        $w_i(l+1) \leftarrow w_i(l)\, w^{\mathsf{i}}_{l+1}\big(\mathbf{x}_i(l+1)\big|\mathbf{x}_i(l)\big)$
    end
end
Output: Weighted ensemble $\big\{(w_i, \mathbf{x}_i), i \in (N_{\mathrm{e}} : 1)\big\}\ [t_0 \rightarrow t_k]$
```

The theory of IS, and in particular theorems 3.3 and 3.4, apply to this case. This means that, in the limit of an infinite ensemble, $N_{\mathrm{e}} \rightarrow \infty$, $\bar{F}_l$ almost surely converges towards F_l, with bias and variance both inversely proportional to the ensemble size N_{e}.

By construction, the particles are independent, which means that the SIS algorithm is embarrassingly parallel. Moreover, since the focus is on the analysis density, it is only necessary to store the particles $\mathbf{x}_{N_{\mathrm{e}}:1}$ and their importance weights $w_{N_{\mathrm{e}}:1}$ at the current time. Therefore, the order of the loops (first over the ensemble, then over time) can be reversed. Hence the algorithmic complexity of the SIS algorithm, is $\mathcal{O}(N_{\mathrm{e}})$ per time step and it can be reduced by a factor N_{t}, the number of threads running in parallel.

A major drawback of the SIS algorithm is that the variance of the importance weight function w, unconditional upon the observations, increases over time (Kong et al. 1994). In practice, after only a few time steps, one particle gets all the weight, and hence a lot of computational effort is devoted to update highly unlikely particles. An empirical fix to this issue is to reset the algorithm by using resampling. This is the basis for particle filtering, as presented in section 3.2.

Remark 11. From the expression of the importance weight function w, equation (3.22), we conclude that in the SIS algorithm, the normalisation constant estimated by the sum of the importance weights $w_{N_{\mathrm{e}}:1}$ is $\pi[\boldsymbol{y}_{k:}](\mathbf{y}_{k:})$.

3.2 The particle filter

The PF is a class of DA algorithms based on SIS and suited for the generic filtering estimation problem, problem 1.1. This section presents the main algorithmic ingredients of the PF.

3.2.1 Principle of the particle filter

In the PF, the knowledge on the system is determined by a weighted ensemble $\big\{(w_i, \mathbf{x}_i) \in \mathbb{R}_+ \times \mathbb{R}^{N_\mathsf{x}}, i \in (N_\mathrm{e} : 1)\big\}$. As in the EnKF, each particle $\mathbf{x}_i$ represents a possible realisation of the state vector $\boldsymbol{x}$. The novelty is that to each particle $\mathbf{x}_i$ is attached an importance weight w_i, proportional to how probable the particle is.

Therefore, the goal of the PF is to recursively construct the sequences $(\mathbf{w}_k)_{k\in\mathbb{N}}$ and $(\mathbf{E}_k)_{k\in\mathbb{N}}$, where, as in the EnKF, $\mathbf{E}$ is the ensemble, and where $\mathbf{w} \in \mathbb{R}^{N_\mathrm{e}}$ is the importance weight vector, that is, the vector whose elements are the importance weights $w_{N_\mathrm{e}:1}$. The decomposition of the assimilation cycle is slightly different from the EnKF. The **sampling step** describes how to update the ensemble, the **importance step** describes how to update the importance weight vector, and the optional **resampling step** resets the algorithm.

3.2.1.1 Initialisation

Following step 1 of algorithm 3.1, the i-th particle is initialised as

$$\mathbf{x}_i(0) \sim \nu^\mathsf{qb}. \tag{3.29}$$

Then, following step 2 of algorithm 3.1, its importance weight is computed as

$$w_i(0) = w_0^\mathsf{i}\big(\mathbf{x}_i(0)\big), \tag{3.30}$$

in order to take into account the discrepancy between the background distribution ν^b and the initial proposal distribution ν^qb. Using the matrix notation, the initialisation is written

$$\mathbf{E}_0 \overset{\text{iid}}{\sim} \nu^\mathsf{qb}, \tag{3.31a}$$

$$\mathbf{w}_0 = w_0^\mathsf{i}\big(\mathbf{E}_0\big). \tag{3.31b}$$

3.2.1.2 The sampling step

Following step 5 of algorithm 3.1, the i-th particle is sampled at time t_{k+1} using the proposal distribution ν^q as

$$\mathbf{x}_i(k+1) \sim \nu_k^\mathsf{q}\big[\mathbf{x}_i(k)\big]. \tag{3.32}$$

Using the matrix notation, the sampling step is written

$$\mathbf{E}_{k+1} \sim \nu_k^\mathsf{q}\big[\mathbf{E}_k\big]. \tag{3.33}$$

3.2.1.3 The importance step

Following step 6 of algorithm 3.1, the importance weight of the i-th particle is updated at time t_{k+1} using the incremental importance weight function w^i as

$$w_i(k+1) = w_i(k)\, w_{k+1}^\mathsf{i}\big(\mathbf{x}_i(k+1)\big). \tag{3.34}$$

Using the matrix notation, the importance step is written

$$\mathbf{w}_{k+1} = \mathbf{w}_k \circ w^{\mathsf{i}}_{k+1}\big(\mathbf{E}_k\big), \tag{3.35}$$

where $\circ$ denotes the element-wise multiplication between vectors.

3.2.1.4 The resampling step

In the PF, the importance step is followed by an optional resampling step. The idea of resampling is to replace particles with low importance weights by particles with high importance, and to reset the algorithm. In its most basic form, resampling can be performed as follows.

1. From the weighted ensemble $(\mathbf{w}, \mathbf{E})$, select a surviving particle $\mathbf{x}^{\mathsf{r}}$. In this operation, the probability of selecting the i-th particle is $\bar{w}_i$.
2. Repeat step 1 N_{e} times to obtain the resampled ensemble $\mathbf{E}^{\mathsf{r}}$.
3. Replace $\mathbf{E}$ by $\mathbf{E}^{\mathsf{r}}$ and reset $\mathbf{w}$ to $\mathbf{1}$.

Using this technique, there is a high probability of removing the particles with low importance weights. More details about the resampling step can be found in section 3.3.

Remark 12. The term *resampling* comes from the fact that we sample here for the second time, which results in an additional layer of approximation, as explained in subsection 3.3.1.

3.2.2 The SIR algorithm

The most famous implementation of the PF is the SIR algorithm (Rubin 1987; Gordon et al. 1993), in which

$$\nu^{\mathsf{qb}} = \nu^{\mathsf{b}}, \tag{3.36a}$$

$$\nu^{\mathsf{q}}_k = \nu^{\mathsf{m}}_k, \tag{3.36b}$$

and resampling is performed at every assimilation cycle. In this case, the initial importance weight vector $\mathbf{w}$ is $\mathbf{1}$ and the incremental weight function w^{i}, defined by equations (3.23e) and (3.23f), simplifies into

$$w^{\mathsf{i}}_k = \pi^{\mathsf{o}}_k, \tag{3.37}$$

because there is no discrepancy between the initial proposal distribution ν^{qb} and the background distribution ν^{b} nor between the proposal distribution ν^{q} and the transition distribution ν^{m}. The SIR algorithm is described by algorithm 3.2, written in matrix notation.

3.2.3 The empirical density

3.2.3.1 An alternative description for the IS estimates

Let $\mathcal{F} : \mathbb{R}^{N_{\mathsf{x}}} \to \mathbb{R}$ be a π^{a}-integrable test function. The expectation F of $\mathcal{F}$ over π^{a} is defined as

$$F_k \triangleq \int \mathcal{F}_k(\mathbf{x}_k)\, \pi^{\mathsf{a}}_k(\mathbf{x}_k)\, \mathrm{d}\mathbf{x}_k, \tag{3.38}$$

Algorithm 3.2: Full assimilation cycle for the SIR algorithm. The resampling step is performed using the multinomial resampling algorithm, algorithm 3.3.

```
Input: $\mathbf{E}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\nu^{\mathsf{m}}\ [t_k \rightarrow t_{k+1}]$, $\pi^{\mathsf{o}}\ [t_{k+1}]$
Sampling
    $\mathbf{E}\ \sim \nu^{\mathsf{m}}[\mathbf{E}]$
Importance
    $\mathbf{w}\ \leftarrow \pi^{\mathsf{o}}(\mathbf{E})$
Resampling
    $\mathbf{E}^{\mathsf{r}} \leftarrow \texttt{Resampling}\big(\mathbf{w}, \mathbf{E}\big)$
    $\mathbf{E}\ \leftarrow \mathbf{E}^{\mathsf{r}}$
Output: $\mathbf{E}\ [t_{k+1}]$
```

In the PF, as in the SIS algorithm, algorithm 3.1, F is approximated by the IS estimate

$$\bar{F}_k = \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_i(k)\, \mathcal{F}_k\big(\mathbf{x}_i(k)\big). \tag{3.39}$$

In other words, the analysis density π^{a} is approximated by the empirical analysis density $\bar{\pi}^{\mathsf{a}}$, defined as

$$\bar{\pi}^{\mathsf{a}}_k(\mathbf{x}_k) \triangleq \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_i(k)\, \delta\big(\mathbf{x}_k - \mathbf{x}_i(k)\big). \tag{3.40}$$

Using a weighted sum of Dirac kernels is a convenient way to write $\bar{\pi}^{\mathsf{a}}$. However, one must keep in mind that equation (3.40) cannot be used to compute a point-wise approximation of π^{a}, but indeed only to compute IS estimates, such as equation (3.39). Therefore, convergence results for the PF, such as the ones presented in section 3.5, should always be understood using a notion of weak convergence.

3.2.3.2 Effect of the prediction operator

Applying the prediction operator $\mathcal{P}$, defined by equation (1.18) in subsection 1.2.1, to the empirical analysis density $\bar{\pi}^{\mathsf{a}}$ yields

$$\mathcal{P}_k(\bar{\pi}^{\mathsf{a}}_k)(\mathbf{x}_{k+1}) = \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_i(k)\, \pi^{\mathsf{m}}_k\big(\mathbf{x}_{k+1}\big|\mathbf{x}_i(k)\big). \tag{3.41}$$

We see immediately that the sampling step of the PF is necessary in order to recover a sum of Dirac kernels. The effect of the sampling step on $\bar{\pi}^{\mathsf{a}}$ can be described by the approximate prediction operator $\bar{\mathcal{P}}$, defined as

$$\bar{\mathcal{P}}_k(\bar{\pi}^{\mathsf{a}}_k)(\mathbf{x}_{k+1}) \triangleq \sum_{i=1}^{N_\mathrm{e}} \bar{w}_i(k) \, \frac{\pi^{\mathsf{m}}_k\big(\mathbf{x}_i(k+1)\big|\mathbf{x}_i(k)\big)}{\pi^{\mathsf{q}}_k\big(\mathbf{x}_i(k+1)\big|\mathbf{x}_i(k)\big)} \, \delta\big(\mathbf{x}_{k+1} - \mathbf{x}_i(k+1)\big). \tag{3.42}$$

The approximate prediction operator $\bar{\mathcal{P}}$ takes into account the discrepancy between the transition distribution ν^{m} and the proposal distribution ν^{q}. Therefore, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the approximate prediction operator $\bar{\mathcal{P}}$ should be equivalent to the prediction operator $\mathcal{P}$.[5] In particular, this means that the empirical forecast density $\bar{\pi}^{\mathsf{f}} \triangleq \bar{\mathcal{P}}(\bar{\pi}^{\mathsf{a}})$ can serve as approximation of the forecast density π^{f}.

Remark 13. In the absence of model error, $\pi^{\mathsf{m}}_k\big(\mathbf{x}_{k+1}\big|\mathbf{x}^{\mathsf{a}}_i(k)\big)$ is a Dirac kernel, and the sampling step is trivial.

3.2.3.3 Effect of the correction operator

Applying the correction operator $\mathcal{C}$, defined by equations (1.19a)–(1.19b) in subsection 1.2.1, to the empirical forecast density $\bar{\pi}^{\mathsf{f}}$ yields

$$\mathcal{C}_{k+1}\big(\bar{\pi}^{\mathsf{f}}_{k+1}\big)\big(\mathbf{x}_{k+1}\big) = \sum_{i=1}^{N_\mathrm{e}} \bar{w}_i(k) \, \frac{w^{\mathsf{i}}_{k+1}\big(\mathbf{x}_i(k+1)\big|\mathbf{x}_i(k)\big)}{\pi[\boldsymbol{y}_{k+1}|\boldsymbol{y}_{k:}](\mathbf{y}_{k+1}|\mathbf{y}_{k:})} \, \delta\big(\mathbf{x}_{k+1} - \mathbf{x}_i(k+1)\big). \tag{3.43}$$

However, the effect of the importance step on $\bar{\pi}^{\mathsf{f}}$ can be described by the approximate correction operator $\bar{\mathcal{C}}$, defined as

$$\bar{\mathcal{C}}_{k+1}\big(\bar{\pi}^{\mathsf{f}}_{k+1}\big)\big(\mathbf{x}_{k+1}\big) \triangleq \bar{\pi}^{\mathsf{a}}_{k+1}\big(\mathbf{x}_{k+1}\big) = \sum_{i=1}^{N_\mathrm{e}} \bar{w}_i(k+1) \, \delta\big(\mathbf{x}_{k+1} - \mathbf{x}_i(k+1)\big). \tag{3.44}$$

In the PF, the importance weights are updated using equation (3.34), and the normalisation constant estimated by the sum of the weights is $\pi[\boldsymbol{y}_{k:}](\mathbf{y}_{k:})$. Therefore, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the approximate correction operator $\bar{\mathcal{C}}$ should be equivalent to the correction operator $\mathcal{C}$.

For completeness, the first assimilation step is described in terms of densities by

$$\bar{\mathcal{C}}_0\big(\pi^{\mathsf{b}}\big)\big(\mathbf{x}_0\big) \triangleq \bar{\pi}^{\mathsf{a}}_0\big(\mathbf{x}_0\big) = \sum_{i=1}^{N_\mathrm{e}} \bar{w}_i(0) \, \delta\big(\mathbf{x}_0 - \mathbf{x}_i(0)\big). \tag{3.45}$$

Remark 14. The only difference between equations (3.43) and (3.44) is that the importance weights in equation (3.43) are in general not normalised. Hence $\mathcal{C}_{k+1}\big(\bar{\pi}^{\mathsf{f}}_{k+1}\big)$ is not rigorously speaking an empirical density. However, the difference asymptotically vanishes because, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, the sum of the weights almost surely converges towards the normalisation constant $\pi[\boldsymbol{y}_{k:}](\mathbf{y}_{k:})$. This demonstrates the limitation of the use of empirical densities.

[5] Read section 3.5 for a presentation of rigorous convergence results.

3.2.3.4 Effect of the resampling step and full assimilation cycle

Without resampling, the PF is described in terms of densities by the recursion

$$\bar{\pi}_0^{\mathsf{a}} = \bar{\mathcal{C}}_0(\pi^{\mathsf{b}}), \tag{3.46a}$$

$$\bar{\pi}_{k+1}^{\mathsf{a}} = \bar{\mathcal{C}}_{k+1} \circ \bar{\mathcal{P}}_k(\bar{\pi}_k^{\mathsf{a}}). \tag{3.46b}$$

Without going into details, let $\bar{\mathcal{R}}$ be the resampling operator, which describes the effect of the resampling step on the empirical densities. In the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$, the resampling operator $\bar{\mathcal{R}}$ should be equivalent to the identity operator. The recursion of the PF then becomes

$$\bar{\pi}_0^{\mathsf{a}} = \bar{\mathcal{R}}_0 \circ \bar{\mathcal{C}}_0(\pi^{\mathsf{b}}), \tag{3.47a}$$

$$\bar{\pi}_{k+1}^{\mathsf{a}} = \bar{\mathcal{R}}_{k+1} \circ \bar{\mathcal{C}}_{k+1} \circ \bar{\mathcal{P}}_k(\bar{\pi}_k^{\mathsf{a}}), \tag{3.47b}$$

as opposed to the recursion described by equations (1.21a) and (1.22) for the analysis density π^{a}. An explicit comparison between both recursions is discussed in section 3.5.

3.3 The resampling step

As mentioned in subsection 3.1.3, after only a few assimilation cycles of the PF without resampling, one particle gets all the weight. This phenomenon is called **weight degeneracy**.[6] The goal of the resampling step is to reinitialise the ensemble: after the resampling step, the ensemble $\mathbf{E}$ is made of N_{e} equally weighted particles. In this section, practical algorithms to implement the resampling step are presented and discussed.

3.3.1 The multinomial and the systematic resampling algorithms

3.3.1.1 Multinomial resampling

The resampling technique described in subsection 3.2.1 is often called *multinomial* resampling, because the selection of particles is equivalent to associating to the i-th analysis particle a number of offspring n_i, such that $n_{N_{\mathrm{e}}:1}$ is a random draw from the multinomial distribution with parameters $(N_{\mathrm{e}}, \bar{\mathbf{w}})$. The multinomial resampling algorithm is described by algorithm 3.3. Smart implementations of this algorithm (*e.g.*, by Ripley 1987) have algorithmic complexity $\mathcal{O}(N_{\mathrm{e}})$.

By construction, the resampling step consists in drawing an iid sample from the distribution whose pdf is the empirical analysis density $\bar{\pi}^{\mathsf{a}}$. Therefore, it is possible to use the resampled ensemble $\mathbf{E}^{\mathsf{r}}$ to approximate $\bar{\pi}^{\mathsf{a}}$ by the resampled empirical density

$$\bar{\bar{\pi}}_k^{\mathsf{a}}(\mathbf{x}_k) = \sum_{i=1}^{N_{\mathrm{e}}} \frac{n_i}{N_{\mathrm{e}}} \delta\big(\mathbf{x}_k - \mathbf{x}_i^{\mathsf{a}}(k)\big). \tag{3.48}$$

[6] Some authors use the term *particle degeneracy*, or *filter degeneracy*. In this thesis, the term *weight degeneracy* is preferred because it is more explicit.

Algorithm 3.3: Multinomial resampling algorithm.

```
Input: Weighted ensemble $(\mathbf{w}, \mathbf{E})$
for $i = 1$ to $N_\mathrm{e}$ do
    $c_i \leftarrow \sum_{j=1}^{i} \bar{w}_j$              // the $i$-th cum. norm. importance weight
end
for $i = 1$ to $N_\mathrm{e}$ do
    $u_i \sim \mathcal{U}[0, 1]$                          // one random draw per particle
    $\psi(i) \leftarrow \min\{j \in (N_\mathrm{e} : 1) \setminus u_i \leq c_j\}$
    $\mathbf{x}_i^\mathsf{r} \leftarrow \mathbf{x}_{\psi(i)}$
end
Output: Resampled ensemble $(\mathbf{1}, \mathbf{E}^\mathsf{r})$
```

By construction, we have

$$\mathbb{E}[n_i] = N_\mathrm{e}\bar{w}_i(k), \tag{3.49}$$

where the expectation operator refers to independent random draws from the multinomial distribution. This means that $\bar{\bar{\pi}}^\mathsf{a}$ is an unbiased approximation of $\bar{\pi}^\mathsf{a}$. However, it can be shown that $\bar{\bar{\pi}}^\mathsf{a}$ yields higher variance IS estimates (see, *e.g.*, Künsch 2005) than $\bar{\pi}^\mathsf{a}$. The extra variance comes here from the random draws from the sampling noise introduced while sampling from the multinomial distribution. As a consequence, resampling can be seen as a way to trade future stability for an increase in immediate variance (Doucet and Johansen 2011).

3.3.1.2 Systematic resampling

Several resampling algorithms have been designed with the goal to reduce the sampling noise. The most popular is probably the systematic resampling algorithm[7] introduced by Kitagawa (1996), and described in algorithm 3.4. Again, smart implementations of this algorithm have algorithmic complexity $\mathcal{O}(N_\mathrm{e})$.

As the multinomial resampling algorithm, the systematic resampling algorithm yields an unbiased approximation of the empirical analysis density $\bar{\pi}^\mathsf{a}$, and it can be shown that it yields the lowest sampling noise over unbiased stochastic resampling algorithms (C. P. Robert and Casella 2004). However, the price to pay for reducing the sampling noise is that we introduce complex dependence between particles, which may be hard to control. This is discussed in section 3.5.

[7] Also known as stochastic universal sampling.

Algorithm 3.4: Systematic resampling algorithm.

```
Input: Weighted ensemble $(\mathbf{w}, \mathbf{E})$
for $i = 1$ to $N_\mathrm{e}$ do
    $c_i \leftarrow \sum_{j=1}^{i} \bar{w}_j$            // the $i$-th cum. norm. importance weight
end
$u \sim \mathcal{U}[0,1]$                                // one random draw for all particles
for $i = 1$ to $N_\mathrm{e}$ do
    $u_i \leftarrow (u + i - 1)/N_\mathrm{e}$
    $\psi(i) \leftarrow \min\{j \in (N_\mathrm{e} : 1) \setminus u_i \leq c_j\}$
    $\mathbf{x}^\mathsf{r}_i \leftarrow \mathbf{x}_{\psi(i)}$
end
Output: Resampled ensemble $(\mathbf{1}, \mathbf{E}^\mathsf{r})$
```

3.3.2 Issues related to resampling

3.3.2.1 Sampling noise and the effective ensemble size

In a PF algorithm, skipping the resampling step means that some computational time is imparted to particles with potentially very low importance weights, whose contribution to the the empirical analysis density $\bar{\pi}^\mathsf{a}$ is very small. On the other hand, both the multinomial and the systematic resampling algorithms introduce sampling noise. Therefore, the choice of the resampling frequency is critical in the design of PF algorithms. Criteria to decide if a resampling step is needed are usually based on measures of the degeneracy, the most popular being probably the effective ensemble size N_eff (Kong et al. 1994), defined for a normalised importance weight vector $\bar{\mathbf{w}}$ by

$$N_\mathrm{eff} \triangleq \left(\bar{\mathbf{w}}^\mathsf{T}\bar{\mathbf{w}}\right)^{-1}. \tag{3.50}$$

If the importance weights are equal, then N_eff is equal to the ensemble size N_e, and if all importance weights but one are null, then N_eff is equal to 1. Therefore, a common strategy is to perform a resampling step if N_eff fall below a fixed threshold, *e.g.*, $N_\mathrm{e}/2$.

The effective ensemble size N_eff can be interpreted as follows. Let $(\boldsymbol{u}_i)_{i\in\mathbb{N}}$ be a sequence of iid random vectors. The random vector $\bar{\boldsymbol{u}}$ defined as

$$\bar{\boldsymbol{u}} \triangleq \sum_{i=1}^{N_\mathrm{e}} w_i \boldsymbol{u}_i, \tag{3.51}$$

has variance given by

$$\mathbb{V}[\bar{\boldsymbol{u}}] = \frac{\mathbb{V}[\boldsymbol{u}]}{N_\mathrm{eff}}. \tag{3.52}$$

If the importance weights $w_{N_\mathrm{e}:1}$ were equal, the variance of $\bar{\boldsymbol{u}}$ would be given by

$$\mathbb{V}[\bar{\boldsymbol{u}}] = \frac{\mathbb{V}[\boldsymbol{u}]}{N_\mathrm{e}}. \tag{3.53}$$

Therefore N_eff is the number of equally-weighted iid samples necessary to yield the same variance reduction as the weighted average $\bar{\boldsymbol{u}}$.

Following the notation of section 3.1.2, the effective ensemble size N_eff can also be approximated by

$$N_\mathrm{eff} \approx \frac{N_\mathrm{e}}{1 + \mathbb{V}\big[w(\boldsymbol{v})\big]}. \tag{3.54}$$

Therefore, it can also be interpreted as a measure of the relative efficiency of IS compared to unbiased MC (Kong et al. 1994).

3.3.2.2 Sample impoverishment and regularisation

During the resampling step, unlikely particles are replaced by duplicates of the most probable particles, which means that there is an unavoidable loss of diversity. In the absence of model error, the diversity is never recovered and, after a few assimilation cycles, it is possible (and highly probable) that the ensemble $\mathbf{E}$ *collapses*.[8] With model error, some diversity is recovered during the sampling step, but there is always a risk that the model error $\boldsymbol{e}^\mathsf{m}$ is not strong enough to counteract the loss of diversity inherent to the resampling step. This phenomenon is known in the statistical literature as *sample impoverishment.*

In order to mitigate the influence of sample impoverishment, a possible approach is to include **regularisation** in the resampling step as follows. Prior to resampling, the empirical analysis density $\bar{\pi}^\mathsf{a}$ is defined as

$$\bar{\pi}^\mathsf{a}_k(\mathbf{x}_k) \triangleq \sum_{i=1}^{N_\mathrm{e}} \bar{w}_i(k)\, \mathcal{K}\big(\mathbf{x}_k - \mathbf{x}_i(k)\big), \tag{3.55}$$

instead of using equation (3.40). The sum of Dirac kernels δ has been replaced here by a sum of kernels $\mathcal{K}$ – to be determined, for example using the kernel density estimation (KDE) theory (Silverman 1986; Musso et al. 2001). Finally, the regularised resampling step consists in drawing an iid sample from the distribution whose pdf is the regularised empirical analysis density $\bar{\pi}^\mathsf{a}$.

From a practical point of view, the regularised resampling step is equivalent to adding a regularisation step after the (non-regularised) resampling step as follows

$$\mathbf{E}_k \leftarrow \mathbf{E}_k + \mathbf{Z}_k, \tag{3.56}$$

where each $\mathbf{Z}$ is an ensemble of random draws from the distributions whose pdf is determined by the kernel $\mathcal{K}$. In some ways this method is similar to additive inflation in EnKF algorithms. It is called **post-regularisation**, because the regularisation is added after the correction (that is, after the importance and the resampling step).

[8] By collapse, it is meant that the ensemble $\mathbf{E}$ is made of N_e copies of the same particle.

An alternative to post-regularisation is to use **pre-regularisation**. In this case, the regularisation given by equation (3.56) is added before the importance step. From a theoretical point of view, pre-regularisation is equivalent to use an additional additive model error $\boldsymbol{e}^{\mathsf{m}}$, because it occurs just after the forecast step.

3.3.3 Alternatives to resampling

In the PF, the resampling step is mandatory to reset the algorithm in case of weight degeneracy. However, resampling introduces additional issues: sampling noise and sample impoverishment. This is why part of the research dedicated to the PF aims at providing alternatives to the resampling step.

3.3.3.1 The ensemble transform particle filter

In both the multinomial and the systematic resampling algorithms, the resampled ensemble $\mathbf{E}^{\mathsf{r}}$ is obtained from the prior ensemble as[9]

$$\mathbf{E}_k^{\mathsf{r}} = \mathbf{E}_k \mathbf{T}_{\mathrm{e}}, \tag{3.57}$$

where $\mathbf{T}_{\mathrm{e}}$ is a transformation matrix in ensemble space, whose coefficients are given by

$$\forall (i,j) \in (N_{\mathrm{e}}\!:\!1)^2, [\mathbf{T}_{\mathrm{e}}]_{j,i} = \begin{cases} 1 & \text{if } \mathbf{x}_i^{\mathsf{r}} \text{ is a copy of } \mathbf{x}_j, \\ 0 & \text{else.} \end{cases} \tag{3.58}$$

In the more general linear ensemble transform (LET) framework (Bishop et al. 2001; Reich and Cotter 2015), the transformation matrix $\mathbf{T}_{\mathrm{e}}$ can have non-negative real coefficients. It is subject to the normalisation constraint

$$\forall j \in (N_{\mathrm{e}}\!:\!1), \quad \sum_{i=1}^{N_{\mathrm{e}}} [\mathbf{T}_{\mathrm{e}}]_{i,j} = 1, \tag{3.59}$$

such that the resampled particles can be interpreted as weighted averages of the analysis particles. The LET matrix $\mathbf{T}_{\mathrm{e}}$ is said to be first-order accurate if it preserves the ensemble mean (Acevedo et al. 2017), that is if

$$\forall i \in (N_{\mathrm{e}}\!:\!1), \quad \sum_{j=1}^{N_{\mathrm{e}}} [\mathbf{T}_{\mathrm{e}}]_{i,j} = N_{\mathrm{e}} \bar{w}_i(k). \tag{3.60}$$

Given the importance weight vector $\mathbf{w}$, let $\mathbb{T}(\mathbf{w})$ be the set of LET matrices with non-negative coefficients satisfying the normalisation constraint, equation (3.59), and the first-order accuracy condition, equation (3.60).

The ensemble transform particle filter (ETPF) algorithm introduced by Reich (2013) is a variant of the SIR algorithm, in which the resampling is replaced by equation (3.57) with $\mathbf{T}_{\mathrm{e}}$

[9] In this section, the terms *prior* and *posterior* refer to the resampling step. Therefore the prior ensemble is the ensemble before resampling and the posterior ensemble is the resampled ensemble.

Algorithm 3.5: Full assimilation cycle for the ETPF algorithm.

```
Input: $\mathbf{E}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\nu^{\mathsf{m}}\ [t_k \to t_{k+1}]$, $\pi^{\mathsf{o}}\ [t_{k+1}]$
Sampling
    $\mathbf{E} \sim \nu^{\mathsf{m}}[\mathbf{E}]$
Importance
    $\mathbf{w} \leftarrow \pi^{\mathsf{o}}(\mathbf{E})$
Resampling
    $\mathbf{T}_{\mathrm{e}}^{*} \leftarrow \underset{\mathbf{T}_{\mathrm{e}} \in \mathbb{T}(\mathbf{w})}{\arg\min}\, \mathcal{J}[\mathbf{E}]\big(\mathbf{T}_{\mathrm{e}}\big)$
    $\mathbf{E} \leftarrow \mathbf{E}\mathbf{T}_{\mathrm{e}}^{*}$
Output: $\mathbf{E}\ [t_{k+1}]$
```

being the *optimal* LET matrix $\mathbf{T}_{\mathrm{e}}^{*}$, defined as the LET matrix minimising the cost function

$$\mathcal{J}[\mathbf{E}_k](\mathbf{T}_{\mathrm{e}}) \triangleq \sum_{i=1}^{N_{\mathrm{e}}} \sum_{j=1}^{N_{\mathrm{e}}} \big[\mathbf{T}_{\mathrm{e}}\big]_{i,j} \big\|\mathbf{x}_i(k) - \mathbf{x}_j(k)\big\|_2^2 \tag{3.61}$$

over all LET matrices in $\mathbb{T}(\mathbf{w})$. This is described by algorithm 3.5.

This minimisation problem is known in the statistical literature as discrete optimal transport (see Villani 2009, and references therein). In this case, $\mathbb{T}(\mathbf{w})$ is interpreted as the set of coupling between the random vectors $\boldsymbol{x}^{\mathsf{f}}$ and $\boldsymbol{x}^{\mathsf{a}}$, defined as the discrete random vectors with realisations in the prior ensemble $\mathbf{x}_{N_{\mathrm{e}}:1}$ and probability vectors $\mathbf{1}/N_{\mathrm{e}}$ for $\boldsymbol{x}^{\mathsf{f}}$ and $\mathbf{w}$ for $\boldsymbol{x}^{\mathsf{a}}$. The cost function defined by equation (3.61) is then equal to the expected distance between $\boldsymbol{x}^{\mathsf{f}}$ and $\boldsymbol{x}^{\mathsf{a}}$. Therefore by construction, the optimal coupling $\mathbf{T}_{\mathrm{e}}^{*}$ minimises the expected distance between the prior and posterior ensemble.

Suppose that the prior ensemble $\mathbf{E}$ is an iid sample from a distribution ν with pdf π. Theorem 3.1, ensures that the empirical density $\bar{\pi}$ of $\mathbf{E}$ weakly converges towards π in the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$. Then, using theorem 1 of Reich (2013), we conclude that, in the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$, the sequence of linear maps $\mathbf{x} \mapsto \mathbf{T}_{\mathrm{e}}^{*}\mathbf{x}$ weakly converges towards a continuous (nonlinear) map $\mathcal{T} : \mathbb{R}^{N_{\mathrm{x}}} \to \mathbb{R}^{N_{\mathrm{x}}}$ such that, if the distribution of a random vector $\boldsymbol{x}$ has pdf π, then the distribution of the random vector $\mathcal{T}(\boldsymbol{x})$ has pdf $\mathcal{C}(\pi)$. Moreover, the expected distance between the random vectors $\boldsymbol{x}$ and $\mathcal{T}(\boldsymbol{x})$ is minimised among such maps.

This result essentially states that the ETPF algorithm satisfies Bayes' theorem in the limit of an infinite ensemble, $N_{\mathrm{e}} \to \infty$. Compared to the multinomial and systematic resampling algorithms, using discrete optimal transport[10] instead of resampling has two major advantages.

[10] Also known as *optimal ensemble coupling*.

First it does not introduce sampling noise. Second, the posterior particles $\mathbf{x}^{\mathsf{r}}_{N_{\mathrm{e}}:1}$ are weighted averages of the prior particles $\mathbf{x}_{N_{\mathrm{e}}:1}$ instead of full copies, which mitigates the sample impoverishment inherent to resampling. Furthermore, as a result of the minimisation, it creates a stronger correlation between the prior and the posterior ensembles $\mathbf{E}$ and $\mathbf{E}^{\mathsf{r}}$ than the multinomial and systematic resampling algorithms. In ensemble DA, this is often an advantage, as shown by the numerical illustration from chapter 5. However, one must keep in mind that this is not always the case. Indeed, as mentioned in paragraph 2.3.2.3, adding random rotations after the analysis step can be beneficial for the performances if deterministic EnKF algorithms.

3.3.3.2 Transport particle filters

Let $\boldsymbol{x}$ be a random vector, whose distribution $\nu[\boldsymbol{x}]$ has pdf $\pi[\boldsymbol{x}]$, and let $\mathcal{T} : \mathbb{R}^{N_{\mathrm{x}}} \to \mathbb{R}^{N_{\mathrm{x}}}$ be an invertible map. The distribution of the random vector $\mathcal{T}(\boldsymbol{x})$ has pdf given by

$$\pi[\mathcal{T}(\boldsymbol{x})] = \mathcal{T}^{\#}\pi[\boldsymbol{x}], \tag{3.62}$$

where $\mathcal{T}^{\#}$ is the pushforward by $\mathcal{T}$, defined for any pdf π over $\mathbb{R}^{N_{\mathrm{x}}}$ by

$$\mathcal{T}^{\#}\pi \triangleq \pi \circ \mathcal{T}^{-1} \cdot \left|\det \mathbf{T}^{-1}\right|, \tag{3.63}$$

in which $\mathbf{T}$ is the Jacobian matrix of the map $\mathcal{T}$.

Suppose that the transport map $\mathcal{T}$ is constructed in such a way that

$$\pi^{\mathsf{a}}_{k} = \mathcal{T}^{\#}_{k}\pi^{\mathsf{f}}_{k}. \tag{3.64}$$

Then, the resampling step at time t_k can be replaced by

$$\forall i \in (N_{\mathrm{e}}\!:\!1), \quad \mathbf{x}^{\mathsf{r}}_{i}(k) = \mathcal{T}_{k}\big(\mathbf{x}_{i}(k)\big). \tag{3.65}$$

For smooth forecast and analysis densities π^{f} and π^{a}, there exists at least one transport map $\mathcal{T}$ satisfying equation (3.64). Following the optimal transport approach, as exposed in the previous paragraph, a possibility is to choose the transport map $\mathcal{T}$ which minimises the cost function

$$\mathcal{J}_{k}(\mathcal{T}_{k}) = \int \left\|\mathbf{x}_{k} - \mathcal{T}_{k}(\mathbf{x}_{k})\right\|_{2}^{2} \pi^{\mathsf{f}}_{k}(\mathbf{x}_{k})\,\mathrm{d}\mathbf{x}_{k}, \tag{3.66}$$

over all maps $\mathcal{T} : \mathbb{R}^{N_{\mathrm{x}}} \to \mathbb{R}^{N_{\mathrm{x}}}$ satisfying equation (3.64). However, when $N_{\mathrm{x}} > 1$, it is even challenging to find only one transport map $\mathcal{T}$ satisfying equation (3.64). Furthermore, π^{f} and π^{a} are in general only known through the empirical forecast and analysis densities $\bar{\pi}^{\mathsf{f}}$ and $\bar{\pi}^{\mathsf{a}}$.

In the applied mathematics community, several approaches emerge to construct an approximate solution to equation (3.64). The central idea is to minimise the Kullback–Leibler divergence between π^{a} and $\mathcal{T}^{\#}\pi^{\mathsf{f}}$ using specific classes of transport maps $\mathcal{T}$.

- In the variational Stein descent method (Liu and Wang 2016), it is assumed that the transport map $\mathcal{T}$ is an element of a reproducing-kernel Hilbert space. The reproducing property is used to derive a simple expression of the functional gradient of the Kullback–Leibler divergence. Pulido and van Leeuwen (2019) apply this method to the PF, the resulting algorithm being called the variational mapping particle filter (VMPF) algorithm.

- Another possibility, proposed by Spantini et al. (2018), is to define the transport map $\mathcal{T}$ using the Knothe–Rosenblatt rearrangement on the state space $\mathbb{R}^{N_\mathsf{x}}$. In practice, the Knothe–Rosenblatt rearrangement is approximated using a sequence of polynomials with increasing degree. This is easy to parametrise and leads to an optimisation problem which can be solved using iterative optimisation methods.

Remark 15. When the proposal distribution ν^q is different from the transition distribution ν^m, the prior ensemble $\mathbf{E}$ is not distributed according to the empirical forecast density $\bar{\pi}^\mathsf{f}$. In this case, equation (3.64) must be modified accordingly.

3.3.3.3 Particle flow particle filters

Consider the stochastic differential equation

$$\mathrm{d}\mathbf{x}_k = \mathcal{A}(\mathbf{x}_k)\,\mathrm{d}\lambda + \mathcal{B}(\mathbf{x}_k)\,\mathrm{d}\epsilon, \tag{3.67}$$

where λ is a pseudo-time, where $\mathcal{A}$ and $\mathcal{B}$ are the drift and diffusion terms (which may depend on the pseudo-time λ), and where ϵ is a Brownian motion. For particles moving according to equation (3.67), the distribution of the associated random variable has density π^λ, determined by the Fokker-Planck equation.

For appropriate values of the drift and diffusion terms $\mathcal{A}$ and $\mathcal{B}$ (see, *e.g.*, Bunch and Godsill 2016), the density π can be obtained as

$$\pi_k^\lambda = \frac{\pi_k^\mathsf{f}\left[\pi_k^\mathsf{o}\right]^\lambda}{\displaystyle\int \pi_k^\mathsf{f}(\mathbf{x}_k)\left[\pi_k^\mathsf{o}(\mathbf{x}_k)\right]^\lambda \mathrm{d}\mathbf{x}_k}. \tag{3.68}$$

which means that $\pi^0 = \pi^\mathsf{f}$ and $\pi^1 = \pi^\mathsf{a}$. As a consequence, instead of defining the transport map in a global sense with equation (3.64), it is possible to use the stochastic differential equation (3.67) in order to move the particles from the forecast distribution ν^f to the analysis distribution ν^a. These principles are used in a variety of PF algorithms, whose implementation differ by the approximation made to compute appropriate values for the drift and diffusion terms $\mathcal{A}$ and $\mathcal{B}$ (see Bunch and Godsill 2016, and references therein).

3.3.3.4 Terminology

In this subsection, several alternatives to resampling have been introduced. From now on, in order to avoid confusion, we distinguish the following cases.

1. If the posterior ensemble is computed with equation (3.57), and if the LET matrix $\mathbf{T}_\mathrm{e}$ has coefficients given by equation (3.58), the update is called, as before, the *resampling* step.

2. If the posterior ensemble is computed with equation (3.57), and if the LET matrix $\mathbf{T}_\mathrm{e}$ has non-negative real coefficients, then the update is called the *linear transformation* step.

Optimal ensemble coupling is a possible implementation of the linear transformation step.

3. If the posterior ensemble is computed with equation (3.65), then the update is called the *transport* step.
4. If the posterior ensemble is computed with the stochastic differential equation, equation (3.67), then the update is called the *particle flow* step.

3.4 The proposal distribution

In this section, we give a brief overview of the use of proposal densities in the PF.

3.4.1 The standard proposal

In the SIR algorithm, we have chosen to use the **standard proposal**, for which the initial proposal distribution ν^{qb} is the background distribution ν^{b} and the proposal distribution ν^{q} is the transition distribution ν^{m}. This corresponds to the proposal vector $\boldsymbol{v} = \boldsymbol{x}$. This choice is convenient because the transition distribution ν^{m} is usually easy to sample from, and because the resulting equation for the incremental weight function w^{i}, equation (3.37), is very simple.

Remark 16. In this case, the importance step is equivalent to a forecast step. Therefore, the importance and resampling steps can be considered as the (equivalent) analysis step.

3.4.2 The optimal importance proposal

As shown in subsection 3.1.2, in terms of efficiency, a good strategy is to use a proposal distribution which minimises the variance of the importance weight function w. A theoretical answer to this problem is to use the **optimal importance proposal** (Kong et al. 1994; Doucet et al. 2000), corresponding to the proposal vector $\boldsymbol{v}$ defined as

$$\boldsymbol{v}_0 \triangleq \boldsymbol{x}_0 | \mathbf{y}_0, \tag{3.69a}$$

$$\boldsymbol{v}_{k+1} \triangleq \boldsymbol{x}_{k+1} | \boldsymbol{x}_k, \mathbf{y}_{k+1}. \tag{3.69b}$$

Therefore, the optimal importance distribution is defined by

$$\nu^{\mathsf{qb}} \triangleq \nu[\boldsymbol{x}_0 | \mathbf{y}_0], \tag{3.70a}$$

$$\nu^{\mathsf{q}}_k \triangleq \nu\big[\boldsymbol{x}_{k+1} | \boldsymbol{x}_k, \mathbf{y}_{k+1}\big]. \tag{3.70b}$$

With this choice, the incremental importance weight function w^{i}, given by equations (3.23e) and (3.23f), simplifies into

$$w^{\mathsf{i}}_0(\mathbf{x}_0) = \pi[\boldsymbol{y}_0](\mathbf{y}_0), \tag{3.71a}$$

$$w^{\mathsf{i}}_{k+1}(\mathbf{x}_{k+1} | \mathbf{x}_k) = \pi[\boldsymbol{y}_{k+1} | \boldsymbol{x}_k](\mathbf{y}_{k+1} | \mathbf{x}_k). \tag{3.71b}$$

It is remarkable that w^{i}_k, the incremental importance weight function at time t_k, does not depend on $\mathbf{x}_k$, the state at time t_k, which means that the variance $\mathbb{V}\big[w^{\mathsf{i}}_k(\boldsymbol{v}_k)\big]$ is null. In other words, the optimal importance distribution nullifies the variance of the importance weight function w_k conditional upon the sequence of previous states $\boldsymbol{x}_{k-1:}$ (defined as an empty set if $k = 0$) and the sequence of observation vectors $\boldsymbol{y}_{k:}$ (Doucet et al. 2000). Furthermore, as shown by Snyder et al. (2015), the optimal importance distribution minimises the variance $\mathbb{V}\big[w_{k:}(\boldsymbol{v}_{k:})\big]$ over all proposal distributions whose pdf can be factored as in equation (3.21). In other words, the optimal importance distribution minimises the variance of the importance weight function w_k conditional only upon the observation vectors $\boldsymbol{y}_{k:}$.

3.4.3 The optimal importance distribution of the GL system

Temporarily assume that the DA system is the GL system, system (1.25). Following Doucet et al. (2000), we define the error covariance matrix $\mathbf{P}$ as

$$\mathbf{P}^{-1} = \mathbf{Q}^{-1} + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}. \tag{3.72}$$

It can be shown that the distribution of the optimal proposal vector $\boldsymbol{v}$ is

$$\boldsymbol{v}_{k+1} \sim \mathcal{N}\Big[\mathbf{P}\big(\mathbf{Q}^{-1}\mathbf{M}\boldsymbol{x}_k + \mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{y}_{k+1}\big), \mathbf{P}\Big], \tag{3.73}$$

and that the incremental weight function is given by

$$w^{\mathsf{i}}_{k+1}(\mathbf{x}_{k+1}|\mathbf{x}_k) \propto \exp\Big[-\frac{1}{2}\big(\mathbf{y}_{k+1} - \mathbf{M}\mathbf{H}\mathbf{x}_k\big)^{\mathsf{T}}\big(\mathbf{H}\mathbf{Q}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}\big(\mathbf{y}_{k+1} - \mathbf{M}\mathbf{H}\mathbf{x}_k\big)\Big]. \tag{3.74}$$

This is very similar to a KF analysis, with the model error covariance matrix $\mathbf{Q}$ playing the role of the forecast error covariance matrix $\mathbf{P}^{\mathsf{f}}$ in the KF.

3.4.4 The proposal distribution in more general DA systems

It turns out that equations (3.72) to (3.74) remain valid if the dynamical model $\mathcal{M}$ is nonlinear. However, in more complex systems there is no analytic formula for the optimal importance distribution and the associated incremental weight function w^{i}, and we need to use more elaborate algorithms to work with the optimal importance distribution, for example the implicit PF algorithm (Chorin and Tu 2009; Chorin et al. 2010; Morzfeld et al. 2012).

To conclude this section, we mention several alternatives to the optimal importance distribution.

- In the auxiliary particle filter algorithm (Pitt and Shephard 1999), the proposal distribution ν^{q} is implicitly defined through the use of an auxiliary member index. The resulting algorithm has two sampling steps: a first one to compute the proposal distribution ν^{q}, and a second one to effectively apply the proposal. This means that the APF algorithm requires $2N_{\mathrm{e}}$ model integrations per time step.
- In the equivalent-weights particle filter (EWPF) algorithm (van Leeuwen 2010; Ades and van Leeuwen 2013), and its implicit version (Zhu et al. 2016), the proposal distribution ν^{q} is computed in such a way that the importance weights of the particles are as uniform as possible after the importance step. The resulting algorithm is quite complex, and dependent on several tuning parameters.

- In the weighted ensemble Kalman filter (WEnKF) algorithm (Papadakis et al. 2010; Morzfeld et al. 2017), the proposal distribution ν^{q} is defined as the assimilation cycle (forecast and analysis step) of the stochastic EnKF algorithm, algorithm 2.2. Obviously, it means that the WEnKF algorithm requires to run the stochastic EnKF algorithm in addition to the PF steps (sampling, importance, and resampling).

In all these algorithms, the proposal distribution ν_k^{q} at time t_k explicitly uses the observation vector $\mathbf{y}_{k+1}$ at time t_{k+1}. This helps the PF sampling particles in regions of the state space $\mathbb{R}^{N_{\mathrm{x}}}$ where the observation density π^{o} is *high*, and indeed contributes to reducing the variance of the importance weight function w. Nevertheless, as mentioned in subsection 3.5.2, and explained in details in section 4.1, this is insufficient to make the PF applicable to high-dimensional DA systems.

3.5 Convergence results for the particle filter

Without resampling, the PF is equivalent to the SIS algorithm, whose convergence results are formalised by theorems 3.3 and 3.4. However, as mentioned in subsection 3.3.1, the resampling step introduces dependence between particles, which means that the PF is outside of the scope of theorems 3.3 and 3.4. Nevertheless, it would be desirable to describe the behaviour of the PF in the limit of an infinite ensemble, $N_{\mathrm{e}} \rightarrow \infty$.

For the SIR algorithm with multinomial resampling, the law of large numbers has been proven by Del Moral (1996) and a central limit theorem has been derived by Del Moral (1999) and Del Moral and Miclo (2000). The central limit theorem has been extended to more general PF algorithms by Chopin (2004) and Künsch (2005), and later by Douc and Moulines (2008). This section presents the most important aspect of the convergence results for PF algorithms.

3.5.1 The law of large numbers for the particle filter

Following Crisan and Doucet (2002), and references therein, to examine the convergence of the empirical analysis density $\bar{\pi}^{\mathsf{a}}$ in the limit of an infinite ensemble, $N_{\mathrm{e}} \rightarrow \infty$, we use the topology of the weak convergence in the space of pdfs over the state space $\mathbb{R}^{N_{\mathrm{x}}}$, defined as follows. The sequence of pdfs $(\pi_{N_{\mathrm{e}}})_{N_{\mathrm{e}}\in\mathbb{N}}$ is said to converge towards the pdf π in the limit $N_{\mathrm{e}} \rightarrow \infty$ if, for any continuous bounded test function $\mathcal{F} : \mathbb{R}^{N_{\mathrm{x}}} \rightarrow \mathbb{R}$,

$$\lim_{N_{\mathrm{e}}\rightarrow\infty} \int \mathcal{F}(\mathbf{x})\, \pi_{N_{\mathrm{e}}}(\mathbf{x})\, \mathrm{d}\mathbf{x} = \int \mathcal{F}(\mathbf{x})\, \pi(\mathbf{x})\, \mathrm{d}\mathbf{x}. \tag{3.75}$$

As deduced from the recursion given by equations (1.21a) and (1.22), the analysis density π_k^{a} at time t_k, with $k > 0$, is given by

$$\pi_k^{\mathsf{a}} = \mathcal{C}_k \circ \mathcal{P}_{k-1} \circ \cdots \circ \mathcal{C}_0\big(\pi^{\mathsf{b}}\big). \tag{3.76}$$

The first condition we want to ensure is that the map $\pi^{\mathsf{b}} \mapsto \pi^{\mathsf{a}}$ is continuous. For this, it is sufficient to check that the observation density π^{o} is continuous bounded and strictly positive, and that the transition density π^{m} is Feller. In other words, for any continuous bounded test function $\mathcal{F} : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$, the map

$$\mathbf{x}_k \mapsto \int \mathcal{F}\big(\mathbf{x}_{k+1}\big)\, \pi^{\mathsf{m}}_k\big(\mathbf{x}_{k+1}|\mathbf{x}_k\big)\, \mathrm{d}\mathbf{x}_{k+1}, \tag{3.77}$$

is continuous bounded as well.

Now, as deduced from the recursion given by equations (3.47a) and (3.47b), the empirical analysis density $\bar{\pi}^{\mathsf{a}}_k$ at time t_k, with $k > 0$, is obtained in the PF as

$$\bar{\pi}^{\mathsf{a}}_k = \bar{\mathcal{R}}_k \circ \bar{\mathcal{C}}_k \circ \bar{\mathcal{P}}_{k-1} \circ \cdots \circ \bar{\mathcal{R}}_0 \circ \bar{\mathcal{C}}_0\big(\pi^{\mathsf{b}}\big). \tag{3.78}$$

The resampling operator $\bar{\mathcal{R}}$ describes a random perturbation, but is has been constructed in such a way that, in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$, $\bar{\mathcal{R}}$ should almost surely converge point-wise towards the identity operator in the space of the pdfs over the state space $\mathbb{R}^{N_\mathrm{x}}$. Similar properties hold for the approximate prediction and correction operators $\bar{\mathcal{P}}$ and $\bar{\mathcal{C}}$.

In general however, the composition of two converging sequence of functions does not converge towards the composition of their limit, unless their convergence satisfies some uniformity condition. Therefore, in order to obtain a law of large numbers for a given PF algorithm, it is sufficient to prove the uniform convergence of the operators $\bar{\mathcal{R}}$, $\bar{\mathcal{P}}$ and $\bar{\mathcal{C}}$. The resulting law of large numbers for the SIR algorithm is formalised in theorem 3.5. This result means that, in a sense, the SIR algorithm provides an asymptotic solution to the generic filtering estimation problem, problem 1.1.

Theorem 3.5 (Law of large numbers for the SIR algorithm)**.** *Under the assumptions that the transition density π^{m} is Feller and that the observation density π^{o} is continuous bounded and strictly positive, the empirical analysis density $\bar{\pi}^{\mathsf{a}}$ of the SIR algorithm almost surely converges towards the analysis density π^{a} of the generic DA system in the limit of an infinite ensemble $N_\mathrm{e} \to \infty$.*

Remark 17. The restriction to continuous bounded test functions $\mathcal{F}$ is necessary to prove the almost sure uniform convergence of the resampling operator $\bar{\mathcal{R}}$. In particular, it excludes the test function $\mathcal{F} : \mathbf{x} \mapsto \mathbf{x}$, which is used to define the mean estimate of the unknown truth $\mathbf{x}^{\mathsf{t}}$. However, in many realistic applications, the truth $\mathbf{x}^{\mathsf{t}}$ always has bounded values, which means that the mean estimate can be defined with a continuous bounded test-function.

3.5.2 The central limit theorem for the particle filter

3.5.2.1 Convergence of the mean squared error

Suppose that the observation density π^{o} is bounded, and let $\mathcal{F} : \mathbb{R}^{N_\mathrm{x}} \to \mathbb{R}$ be a bounded π^{a}-integrable test function. The expectation F of $\mathcal{F}$ under π^{a}, and the IS estimate $\bar{F}$ are defined by equations (3.38) and (3.39).

In this case, using an induction it can be shown (Crisan and Doucet 2002) for the SIR algorithm that

$$\mathbb{E}\Big[\big(\bar{F}_k - F_k\big)^2\Big] \leq \frac{\alpha_k \|\mathcal{F}\|_2^2}{N_\mathrm{e}}, \tag{3.79}$$

where the expectation operator refers to independent random draw for the IS estimate $\bar{F}$, and where the convergence constant α does not depend on the ensemble size N_e. However, α may depend on the dimension of the state space N_x and, worse, it increases over time. From a practical point of view, this means that, in order to ensure a given level of precision, N_e has to increase with time.

Without going into details, we conclude this paragraph by mentioning the fact that, if the transition distribution satisfies some ergodic properties (not detailed here), then is is possible to prove that α is bounded in time. This shows why, in specific cases, the SIR algorithm *works*.

3.5.2.2 The central limit theorem

The central limit theorem for the SIR algorithm shares many aspects with the convergence of the mean squared error, as presented in the previous paragraph. It is stated, in a minimalistic way, in theorem 3.6.

Theorem 3.6 (Central limit theorem for the SIR algorithm)**.** *Under minimal assumptions, the rescaled error $\sqrt{N_\mathrm{e}}\big(\bar{F}-F\big)$ of the SIR algorithm converges in distribution towards $\mathcal{N}\big[0,\sigma^2\big]$ in the limit of an infinite ensemble, $N_\mathrm{e}\rightarrow\infty$, where the asymptotic variance σ^2 is computed by induction over the time index k.*

In theory, the central limit theorem can be used to discriminates between different PF algorithms, the one with the lowest asymptotic variance σ^2 being considered as more efficient. However for complex algorithms, with several layers of approximations, it may be really hard to derive a central limit theorem.

3.5.2.3 The curse of dimensionality

These convergence results – law of large numbers and central limit theorem – are the main reasons for the interest of the DA community in the PF. In terms of application, the story is totally different. Indeed, even in low-order DA systems, any algorithm based on IS requires an extremely high number of particles N_e to yield estimates with a reasonable variance (van Leeuwen 2003; Zhou et al. 2006; van Leeuwen 2009; Bocquet et al. 2010). This is related to the fact that the asymptotic variance σ^2 in the central limit theorem, for example theorem 3.6, may increase exponentially with the dimension of the state space N_x. This point is explained in details in section 4.1.

Part II

Localisation in the particle filter

4 Localisation in the particle filter: methodological aspects

Contents

Despite the keen interest of the DA community in the PF, the conclusion of chapter 3 is firm: even in low-order DA system, a direct application of the PF requires an extremely high number of particles to yield accurate estimates (van Leeuwen 2003; Zhou et al. 2006; van Leeuwen 2009; Bocquet et al. 2010). This is a symptom of the curse of dimensionality and the main obstacle to an application of the PF to most DA systems (Silverman 1986; Kong et al. 1994; Snyder et al. 2008).

In most DA systems, distant regions have an (almost) independent evolution over short timescales. This idea is used in the EnKF to implement localisation. In the PF, localisation could be used to counteract the curse of dimensionality. Yet, if localisation in the EnKF is simple and leads to efficient algorithms, implementing localisation in the PF is a challenge, because there is no trivial way of gluing together locally updated particles across domains (van Leeuwen 2009).

This chapter focuses on the methodological aspects of the implementation of localisation in the PF, as developed by Farchi and Bocquet (2018). The objective is here to review and compare recent propositions of local particle filter (LPF) algorithms and to suggest practical solutions to the difficulties of local particle filtering which lead to improvements in the design of the LPF algorithms. Section 4.1 is dedicated to the curse of dimensionality, with some theoretical elements and illustrations. The challenges of localisation in the PF are then discussed in sections 4.2 and 4.3 from two different point of view. For both approaches, new implementations of LPF algorithms are proposed. Conclusions are given in section 4.5.

4.1 The curse of dimensionality

In the statistical literature, it is a well-known fact that the estimation of continuous pdfs suffers from the curse of dimensionality, meaning that the computational cost increases exponentially with the dimension (Silverman 1986). In this section, the IS method derived in subsection 3.1.2 is shown to face similar limitations, which is an obstacle to the application of the PF to high-dimensional DA systems.

4.1.1 Illustrations of the curse of dimensionality

We start this section by illustrating some pathological behaviour of the IS method when the dimension of the state space N_x increases.

4.1.1.1 The variance of the IS estimate

Consider an N_x-dimensional diagonal generalisation of the simple example presented in subsection 3.1.2. The background and observation densities π^b and π^o are are given by

$$\pi^\mathsf{b}(\mathbf{x}) = \frac{1}{\sqrt{2\pi}^{N_\mathrm{x}}} \exp\left[-\frac{1}{2}\left\|\mathbf{x}-\mathbf{x}^\mathsf{b}\right\|_2^2\right], \tag{4.1a}$$

$$\pi^\mathsf{o}(\mathbf{y}|\mathbf{x}) = \frac{1}{\sqrt{2\pi}^{N_\mathrm{x}}} \exp\left[-\frac{1}{2}\left\|\mathbf{x}-\mathbf{y}\right\|_2^2\right], \tag{4.1b}$$

with $\mathbf{x}^\mathsf{b} = -\mathbf{1}$ and $\mathbf{y} = \mathbf{1}$, and the analysis density π^a, is given by

$$\pi^\mathsf{a}(\mathbf{x}|\mathbf{y}) = \frac{1}{\sqrt{\pi}^{N_\mathrm{x}}} \exp\left[-\left\|\mathbf{x}-\mathbf{x}^\mathsf{a}\right\|_2^2\right], \tag{4.1c}$$

with $\mathbf{x}^\mathsf{a} = \mathbf{0}$. The IS method is used with the standard proposal (that is, with $\nu^\mathsf{q} = \nu^\mathsf{b}$) to estimate the test functions $\mathcal{F}_n : \mathbf{x} \mapsto [\mathbf{x}]_n, n \in (N_\mathrm{x}\!:\!1)$, whose expected values are $F_n = [\mathbf{x}^\mathsf{a}]_n = 0$.

For each $\mathcal{F}_n$, the asymptotic rescaled bias b_∞ and the asymptotic rescaled variance σ^2, given by equations (3.14b) (3.15), are equal to

$$-b_\infty = \frac{1}{3}\exp\left[N_\mathrm{x}\left(\frac{2}{3}+\ln\frac{2\sqrt{3}}{3}\right)\right], \tag{4.2}$$

$$\sigma^2 = \frac{4}{9}\exp\left[N_\mathrm{x}\left(\frac{2}{3}+\ln\frac{2\sqrt{3}}{3}\right)\right]. \tag{4.3}$$

In the one-dimensional case, $N_\mathrm{x}=1$, we recover equations (3.18) and (3.19). Moreover, we immediately see that both quantities increase exponentially with the dimension of the state space N_x. As a consequence, in this simple DA system, in order to maintain a unit variance for $\bar{F}_1$, the IS estimate of only *one* element of $\mathbf{x}^\mathsf{a}$, the number of particles N_e must increase exponentially with N_x.

Furthermore, the dimensions of this DA problem are by construction iid. This means that the strong law of large numbers ensures the almost sure convergence

$$\lim_{N_\mathrm{x}\to\infty}\frac{1}{N_\mathrm{x}}\sum_{n=1}^{N_\mathrm{x}}\left(\bar{F}_n-F_n\right)^2 = \lim_{N_\mathrm{x}\to\infty}\frac{1}{N_\mathrm{x}}\sum_{n=1}^{N_\mathrm{x}}\bar{F}_n^2 = \mathbb{E}\left[\bar{F}_1^2\right] = \mathbb{V}\left[\bar{F}_1\right]+\mathbb{E}\left[\bar{F}_1\right]^2, \tag{4.4}$$

where the expectation and variance operators refer to independent random random draws of the ensemble E according to the background distribution ν^b. Using the convergence of the rescaled bias $N_\mathrm{e}\left(\mathbb{E}\left[\bar{F}_1\right]-F_1\right)$ towards b_∞ and the convergence in distribution of the rescaled error $\sqrt{N_\mathrm{e}}\left(\bar{F}_1-F_1\right)$ towards $\mathcal{N}\left[0,\sigma^2\right]$, we conclude that

$$\mathbb{E}\left[\bar{F}_1^2\right] \underset{N_\mathrm{e}\to\infty}{\sim} \frac{\sigma^2}{N_\mathrm{e}}. \tag{4.5}$$

In other words, when both the dimension of state space N_x and the ensemble size N_e are large, the squared error of the IS estimate of $\mathbf{x}^\mathsf{a}$ is

$$\sum_{n=1}^{N_\mathrm{x}}\left(\bar{F}_n-F_n\right)^2 \approx \frac{\sigma^2 N_\mathrm{x}}{N_\mathrm{e}}, \tag{4.6}$$

where σ^2 increases exponentially with N_x. For comparison, the squared $\mathcal{L}^2$-norm between $\mathbf{x}^\mathsf{b}$ and $\mathbf{x}^\mathsf{a}$ is equal to N_x, and the squared $\mathcal{L}^2$-norm between $\mathbf{y}$ and $\mathbf{x}^\mathsf{a}$ is equal to N_x as well. Therefore, unless N_e is of the same order as $\exp N_\mathrm{x}$, the IS estimate of $\mathbf{x}^\mathsf{a}$ is likely to have *larger* errors than both $\mathbf{x}^\mathsf{b}$ and $\mathbf{y}$. This provides a theoretical formalisation of the numerical results presented in section 3 of Snyder et al. (2008).

4.1.1.2 The weight degeneracy of IS

As explained in subsection 3.1.2, the variance of the importance weights $w_{N_\mathrm{e}:1}$ can be interpreted as a measure of the relative efficiency of the IS method, compared to the MC method. However, in most applications, when the dimension of the state space N_x increases, the occurrences of weight degeneracy become dramatically more frequent.

This phenomenon can be illustrated in the simple DA system of the previous paragraph.

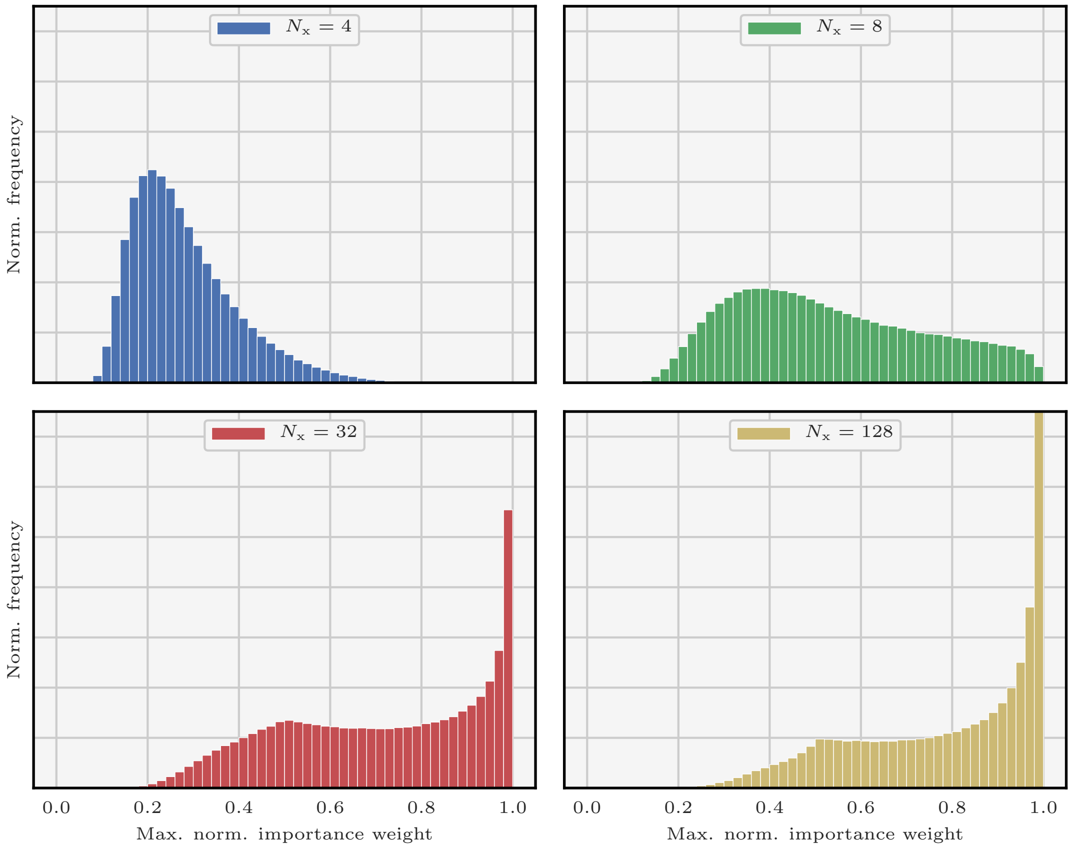

Figure 4.1: Empirical frequencies of the maximum of the normalised importance weights, $\max\{\bar{w}_i, i \in (N_e : 1)\}$, for $N_x = 4$ (top-left panel, in blue), 8 (top-right panel, in green), 32 (bottom-left panel, in red), and 128 (bottom-right panel, in yellow). In all cases, the ensemble size is $N_e = 128$, and the frequencies are computed using 10^7 independent random draws of the ensemble E from the background distribution ν^{b}.

For example, figure 4.1 shows the empirical frequencies of the maximum of the normalised importance weights $\bar{w}_{N_e:1}$, for a fixed ensemble size $N_e = 128$ and several values of N_x. When the number of state variables is small ($N_x = 4$), the $\bar{w}_{N_e:1}$ are balanced, and values close to 1 are infrequent. However, when the number of variables grows ($N_x \geq 32$) the $\bar{w}_{N_e:1}$ rapidly degenerate: values close to 1 become more frequent. Ultimately, the frequency peaks to 1, meaning that, most of the time, only *one* particle has a non-zero contribution to the empirical analysis density $\bar{\pi}^{\mathsf{a}}$, and the IS estimates cannot be expected to be accurate. Similar properties have been observed by Snyder et al. (2008), and by Bocquet et al. (2010) in a more elaborate DA system.

4.1.2 The equivalent state dimension

In the DA system described in the previous subsection, when the dimension of the state space N_x increases, the background and analysis distributions ν^b and ν^a become increasingly singular to each other: random particles drawn from ν^b have an exponentially small likelihood according to π^a. This is the main reason for the drastic increase in the number of particles required for a non-degenerate scenario in the IS method (Rebeschini and van Handel 2015).

A quantitative description of the behaviour of the importance weights $w_{N_\mathrm{e}:1}$ for large values of the number of observations N_y has been proposed by Snyder et al. (2008). In this study, the authors take the example of a single IS step with the the standard proposal ($\nu^\mathsf{q} = \nu^\mathsf{b}$) when the observations are iid. Let $\boldsymbol{s}$ be the random variable defined as

$$\boldsymbol{s} \triangleq \frac{\ln \pi[\boldsymbol{y}|\boldsymbol{x}] - \mu}{\tau}, \tag{4.7a}$$

where μ and τ^2 are given by

$$\mu \triangleq \mathbb{E}\big[\ln \pi[\boldsymbol{y}|\boldsymbol{x}]\big], \tag{4.7b}$$

$$\tau^2 \triangleq \mathbb{V}\big[\ln \pi[\boldsymbol{y}|\boldsymbol{x}]\big]. \tag{4.7c}$$

The random variable $\boldsymbol{s}$ can also be written

$$\boldsymbol{s} = \frac{1}{\tau}\left[\sum_{q=1}^{N_\mathrm{y}} \ln \pi[\boldsymbol{y}_q|\boldsymbol{x}] - \mu\right]. \tag{4.8}$$

which is a sum of iid random variables because the observations are iid. Therefore, the central limit theorem ensures that, in the limit of an infinite number of observation, $N_\mathrm{y} \rightarrow \infty$, $\boldsymbol{s}$ converges in distribution towards $\mathcal{N}[0, 1]$. Moreover, the observation density π^o can be written

$$\pi^\mathsf{o}(\mathbf{y}|\mathbf{x}) = \exp(-\mu - \tau\boldsymbol{s})(\mathbf{y}|\mathbf{x}). \tag{4.9}$$

Using equation (4.9), as well as the convergence in distribution of $\boldsymbol{s}$ towards $\mathcal{N}[0, 1]$, Snyder et al. (2008) have shown, under minimal conditions, that

$$\mathbb{E}\left[\left[\max_{i\in(N_\mathrm{e}:1)} \bar{w}_i\right]^{-1}\right] \underset{N_\mathrm{e}\rightarrow\infty}{\sim} 1 + \frac{\sqrt{2\ln N_\mathrm{e}}}{\tau}, \tag{4.10}$$

where the expectation operator refers to independent random draws of the ensemble E from ν^b.

This result means that, in order to avoid weight degeneracy in an algorithm based on the IS method, N_e must be of order $\exp\big(\tau^2/2\big)$. In simple cases, such as the one considered in the previous subsection, τ^2 is proportional to the number of observations N_y. The dependence of τ^2 on the dimension of the state space N_x is indirect in the sense that the derivation of equation (4.10) requires N_x to be asymptotically large. In a sense, one can think of τ^2 as an **equivalent state dimension** for the DA system.

4.1.3 Mitigating the degeneracy using a proposal distribution

One objective of using proposal densities in the PF is to improve the quality of the IS estimates by reducing the variance of the w function, as discussed in section 3.4. For example, when using the optimal importance distribution, the resulting w function does not depend on the current state. For a single IS step, this means that the importance weights $w_{N_\mathrm{e}:1}$ are all equal.

However, using the optimal importance distribution in a cycled DA system still yields weight degeneracy, as illustrated, *e.g.*, by Bocquet et al. (2010). In this case, the degeneracy does not primarily come from the IS step, but from the recursion in the PF. In particular, it stems from the fact that the PF does not correct the particles at earlier times to account for new observations. This has been a key element in the development of the guided SIR algorithm of van Leeuwen (2009), whose ideas were included in the practical implementations of the EWPF algorithm (van Leeuwen 2010; Ades and van Leeuwen 2013) as a relaxation step.

The theoretical analysis of the previous subsection can be extended to the case of IS with a non-standard proposal distribution by considering the following DA system:

$$\boldsymbol{x}_0 \sim \nu[\boldsymbol{x}_0], \tag{4.11a}$$

$$\boldsymbol{x}_1 = \mathcal{M}(\boldsymbol{x}_0) + \boldsymbol{e}_0^{\mathsf{m}}, \qquad \boldsymbol{e}_0^{\mathsf{m}} \sim \nu\big[\boldsymbol{x}_1 - \mathcal{M}(\boldsymbol{x}_0)\big]. \tag{4.11b}$$

$$\boldsymbol{y}_1 = \mathcal{H}(\boldsymbol{x}_1) + \boldsymbol{e}_1^{\mathsf{o}}, \qquad \boldsymbol{e}_1^{\mathsf{o}} \sim \nu\big[\boldsymbol{y}_1 - \mathcal{H}(\boldsymbol{x}_1)\big]. \tag{4.11c}$$

For this system, the proposal vector $\boldsymbol{v}$ of the standard proposal is distributed as

$$\nu[\boldsymbol{v}_0] \triangleq \nu[\boldsymbol{x}_0], \tag{4.12a}$$

$$\nu[\boldsymbol{v}_1] \triangleq \nu[\boldsymbol{x}_1|\boldsymbol{x}_0], \tag{4.12b}$$

while the proposal vector $\boldsymbol{v}^*$ of the optimal importance proposal is distributed as

$$\nu[\boldsymbol{v}_0^*] \triangleq \nu[\boldsymbol{x}_0], \tag{4.13a}$$

$$\nu[\boldsymbol{v}_1^*] \triangleq \nu[\boldsymbol{x}_1|\boldsymbol{x}_0, \mathbf{y}_1]. \tag{4.13b}$$

Therefore, the importance weight function is given by

$$w(\mathbf{x}_0, \mathbf{x}_1) = \begin{cases} \pi[\boldsymbol{y}_1|\boldsymbol{x}_1](\mathbf{y}_1|\mathbf{x}_1) & \text{for the standard proposal,} \\ \pi[\boldsymbol{y}_1|\boldsymbol{x}_0](\mathbf{y}_1|\mathbf{x}_0) & \text{for the optimal importance proposal.} \end{cases} \tag{4.14}$$

As shown by Snyder et al. (2015), the asymptotic relationship given by equation (4.10) remains valid if the equivalent state dimension τ^2 is defined as

$$\tau^2 = \begin{cases} \mathbb{V}\big[\pi[\boldsymbol{y}_1|\boldsymbol{x}_1]\big] & \text{for the standard proposal,} \\ \mathbb{V}\big[\pi[\boldsymbol{y}_1|\boldsymbol{x}_0]\big] & \text{for the optimal importance proposal.} \end{cases} \tag{4.15}$$

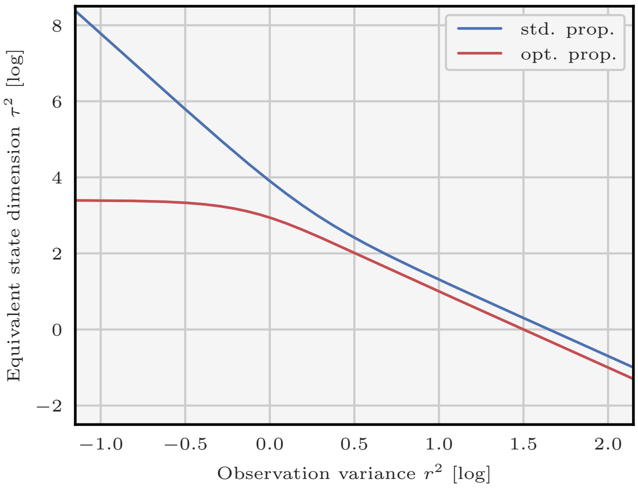

Figure 4.2: Evolution of the equivalent state dimension τ^2 of the standard proposal (in blue) and of the optimal importance proposal (in red) as a function of the observation variance r^2. The other parameters are fixed and equal to $N_\mathrm{x} = 10^3$, $a = p = h = q = 1$.

In the simple example where the DA system is given by

$$\begin{aligned} \boldsymbol{x}_0 &\sim \mathcal{N}\big[\mathbf{0}, p^2\mathbf{I}\big], && && (4.16a) \\ \boldsymbol{x}_1 &= a\boldsymbol{x}_0 + \boldsymbol{e}_0^\mathsf{m}, && \boldsymbol{e}_0^\mathsf{m} \sim \mathcal{N}\big[\mathbf{0}, q^2\mathbf{I}\big]. && (4.16b) \\ \boldsymbol{y}_1 &= h\boldsymbol{x}_1 + \boldsymbol{e}_1^\mathsf{o}, && \boldsymbol{e}_1^\mathsf{o} \sim \mathcal{N}\big[\mathbf{0}, r^2\mathbf{I}\big], && (4.16c) \end{aligned}$$

the equivalent state dimension τ^2 is equal to

$$\tau^2 = \begin{cases} N_\mathrm{x}\dfrac{h^2\big(q^2 + a^2p^2\big)}{r^2}\left[1 + \dfrac{3h^2}{2r^2}\big(q^2 + a^2p^2\big)\right] & \text{for the standard proposal,} \\ N_\mathrm{x}\dfrac{a^2p^2h^2}{r^2 + h^2q^2}\left[1 + \dfrac{3a^2h^2p^2}{2(r^2 + h^2q^2)}\right] & \text{for the optimal proposal.} \end{cases} \tag{4.17}$$

Figure 4.2 shows the evolution of the equivalent state dimension τ^2 as a function of the observation variance r^2 in this DA system. As expected, τ^2 is systematically smaller when using the optimal importance proposal, and much smaller when r^2 is small (typically $r^2 \leq 1$). In both cases however, τ^2 is proportional to the dimension of the state space N_x, which means that ultimately the optimal importance proposal cannot counteract the curse of dimensionality in this simple model, and there is no reason to think that it could in more elaborate models (see chapter 29 of MacKay 2003).

Furthermore, as shown by Snyder et al. (2015), the optimal importance proposal yields the lowest equivalent state dimension τ^2 over single-step proposals, which includes all the algorithms mentioned in subsection 3.4.4.

Remark 18. In the simple DA system described in this subsection, τ^2 is directly proportional to N_{x} (equal to N_{y} in this case). For more elaborate models, the relationship between τ^2 and N_{x} is likely to be more complex and may involve the effective number of degrees of freedom in the model.

4.1.4 Using localisation to avoid the degeneracy

The simple DA systems introduced in subsections 4.1.1 and 4.1.3 are separable, meaning that the state variables independent from each other. By contrast, the IS method operate at a global scale and cannot exploit the separability property. Indeed, the importance weight vector $\mathbf{w}$ is influenced by all elements of the observation vector $\mathbf{y}$, which in turn influences all elements of the IS estimates (for example $\mathbf{x}^{\mathsf{a}}$). This explains why the ensemble size N_{e} must increase exponentially with the dimension of the state space N_{x} to avoid the weight degeneracy. Furthermore, it is easy to understand that the weight degeneracy can be avoided by splitting the global DA system into a collection of N_{x} local DA systems.

Most geophysical systems are not separable. However, the correlations decrease at a fast rate with the distance in the physical space. This is the basis for the implementation of localisation in the EnKF, as presented in section 2.5. Similar principles could be applied to the PF. For example, by considering the counterpart of DL presented in subsection 2.5.4, the influence of each observation would be restricted to a spatial neighbourhood of its site. As a consequence, the equivalent state dimension τ^2 would be defined using the maximum number of observations in each local domain N_{y}^{ℓ}, which could be kept relatively small even for high-dimensional systems. The application of DL in the PF is discussed by Snyder et al. (2008), van Leeuwen (2009) and Bocquet et al. (2010), with an emphasis on two major difficulties.

The first issue is that the variation of the weights across local domains irredeemably breaks the structure of the global particles. There is no trivial way of recovering this global structure, *i.e.*, gluing together the locally updated particles. Yet, global particles are required for the sampling step in the next assimilation cycle of the PF, where the dynamical model $\mathcal{M}$ is applied to each individual particle.

Second, if not carefully constructed, the gluing could yield imbalance, as presented in paragraph 2.5.4.2. In EnKF algorithms using DL, these issues are mitigated by using smooth functions to taper the influence of the observations (*e.g.*, the GC function G). By contrast, in most PF algorithms, this may not be sufficient to avoid some imbalance. This is discussed in details in subsection 4.2.3.

From now on we suppose, as in subsection 2.5.4, that all observations are local. In the following sections, we present different methods which address these two issues, and lead to practical implementations of DL in the PF. The resulting LPF algorithms are divided into two classes. In the first class, independent updates are performed for each state variable by using only the observations influencing the considered variable. This leads to algorithms easy to define, to implement, and to parallelise. However, there is by construction no obvious relationship between state variables, which could yield imbalance. This approach is used for example by Rebeschini and van Handel (2015), Cheng and Reich (2015), Penny and Miyoshi (2016) and Lee and Majda (2016). In the second approach, an update is performed for each observation, with the constraint that only nearby grid points are updated. Within this formalism, the observations must be assimilated sequentially. As a consequence, the resulting algorithms are slightly harder to define and to parallelise, but they may mitigate the imbalance. This approach is used for example by Poterjoy (2016).

4.2 The LPF–X algorithms: a block localisation framework

The LPF algorithms described in this section are constructed as a local generalisation of the SIR algorithm:

1. the sampling step is performed with an equally weighted ensemble $\mathbf{E}$, using the standard proposal;

2. the importance and resampling steps are performed independently for each state variable $n \in (N_\mathrm{x} : 1)$.

Therefore, the sampling step is equivalent to a forecast step,[1] and the (local) importance and sampling steps can be considered as the (equivalent) analysis step. Such LPF algorithms are called LPF–X algorithms, where the –X extension emphasises the fact that there is one update per state variable.

In this section, the terms *prior* and *posterior* (or *updated*) refers to quantities before and after the analysis step of the LPF–X algorithms. For simplicity, the time subscript k is systematically dropped, and the conditioning with respect to prior quantities is implicit. The prior and posterior ensembles are written $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$. In order to avoid confusion between ensemble and state variable indices, we use a functional notation for the ensemble indices and a subscript notation for the state variable indices. With this convention, the i-th particle in an ensemble $\mathbf{E}$ is written $\mathbf{x}(i)$, the n-th element of a vector $\mathbf{x} \in \mathbb{R}^{N_\mathrm{x}}$ is written x_n, and the i-th element of a vector $\mathbf{w} \in \mathbb{R}^{N_\mathrm{e}}$ is written $w(i)$. For example, $x^\mathsf{f}_n(i)$ and $x^\mathsf{a}_n(i)$ are the n-th state variable of the i-th particle in the forecast and analysis ensembles $\mathbf{E}^\mathsf{f}$ and $\mathbf{E}^\mathsf{a}$, and $w_n(i)$ is the local importance weight of the i-th particle for the n-th state variable, as introduced in the following subsection.

In subsection 4.2.1, we show how localisation is introduced in the PF. The generalisation to a block localisation framework is presented in subsection 4.2.2, in which the generic LPF–X algorithm is derived. Finally, the resampling step is explained in details in subsection 4.2.3.

4.2.1 Introducing localisation in the PF

Localisation is generally introduced in the PF by allowing the importance weight vector $\mathbf{w}$, computed during the importance step, to depend on the spatial position. In the (global) PF, the n-th marginal $\bar{\pi}^\mathsf{a}_n$ of the empirical analysis density $\bar{\pi}^\mathsf{a}$ is

$$\bar{\pi}^\mathsf{a}_n(x_n|\mathbf{y}) = \sum_{i=1}^{N_\mathrm{e}} \bar{w}(i)\, \delta\big(x_n - x^\mathsf{f}_n(i)\big), \tag{4.18}$$

[1] In other words, the ensemble $\mathbf{E}$ resulting from the sampling step is distributed according to the forecast distribution ν^f, which is why it is written $\mathbf{E}^\mathsf{f}$.

whose local variant is

$$\bar{\pi}^{\mathsf{a}}_{n}(x_n|\mathbf{y}) = \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_n(i)\,\delta\big(x_n - x^{\mathsf{f}}_n(i)\big), \tag{4.19}$$

where the local importance weight vector $\mathbf{w}_n$ now depends on the spatial position through the state variable index n. Using local importance weight vectors results in uncoupled marginals for $\bar{\pi}^{\mathsf{a}}$. This is the reason why localisation was introduced in the first place, but as a drawback, the full empirical analysis density $\bar{\pi}^{\mathsf{a}}$ is not known. The simplest fix is to approximate the full $\bar{\pi}^{\mathsf{a}}$ as the product of its marginals $\bar{\pi}^{\mathsf{a}}_{N_{\mathrm{x}}:1}$, which is written

$$\bar{\pi}^{\mathsf{a}}(\mathbf{x}|\mathbf{y}) = \prod_{n=1}^{N_{\mathrm{x}}} \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_n(i)\,\delta\big(x_n - x^{\mathsf{f}}_n(i)\big). \tag{4.20}$$

This is a weighted sum of the $N_{\mathrm{e}}^{N_{\mathrm{x}}}$ possible combinations between all particles in the prior ensemble $\mathbf{E}^{\mathsf{f}}$. Resampling from the distribution whose pdf is given by equation (4.20), is equivalent to perform N_{x} independent resampling from the distributions whose pdfs are given by equation (4.19). In other words, the resampling step has to be performed independently for each state variable $n \in (N_{\mathrm{x}}\!:\!1)$, and then the global posterior particles $\mathbf{x}^{\mathsf{a}}(N_{\mathrm{e}}\!:\!1)$ are obtained by assembling the locally resampled particles $x^{\mathsf{r}}_{N_{\mathrm{x}}:1}(N_{\mathrm{e}}\!:\!1)$.

4.2.2 The block localisation framework

4.2.2.1 Generalisation to the block localisation framework

The localisation described in the previous subsection can be embedded into a more general block localisation framework as follows. The state space $\mathbb{R}^{N_{\mathrm{x}}}$ is divided into **local** (state) **blocks** with the additional constraint that the local importance weight vectors $\mathbf{w}_{N_{\mathrm{x}}:1}$ should be constant over the local blocks. As a consequence, the resampling step can be performed *independently for each block*. Typically, the local blocks could be defined as the set of state variables corresponding to a batch of adjacent grid points.

In the block PF algorithm of Rebeschini and van Handel (2015), the local importance weight vector $\mathbf{w}_b$ corresponding to the b-th local block is computed using the observations whose site is located within this block. However, in general, nothing prevents one from using the observations whose site is located within a **local domain** potentially different from the local block. This is the case in the LPF algorithm of Penny and Miyoshi (2016), in which the local blocks have for size 1 grid point, while the size of the local domains is controlled by a parameter.

To summarise, the LPF–X algorithms are characterised by

- the geometry of the local blocks over which the local importance weight vectors $\mathbf{w}_{N_{\mathrm{x}}:1}$ are constant;
- the local domain of each local block, which gathers all observations used to compute the $\mathbf{w}_{N_{\mathrm{x}}:1}$;
- the resampling algorithm.

Most LPF algorithms (*e.g.*, those described in Rebeschini and van Handel 2015; Penny and Miyoshi 2016; Lee and Majda 2016) in the literature can be seen to adopt this block localisation framework.

Remark 19. The concept of local domain, as defined in this subsection, is totally compatible with the notion of local domains, as defined in paragraph 2.5.4.1 for the EnKF algorithms using DL.

4.2.2.2 The local state blocks

Using parallelepipedic local blocks is a standard geometric choice (Rebeschini and van Handel 2015; Penny and Miyoshi 2016). It is easy to conceive and to implement, and it offers a potentially interesting degree of freedom: the shape of the local blocks. Using larger local blocks decreases the proportion of block boundaries, and hence it decreases the bias of the analysis step of the LPF–X algorithms. On the other hand, using large local blocks also means less freedom to counteract the curse of dimensionality.

In the clustered PF algorithms introduced by Lee and Majda (2016), the local blocks are centred around the observation sites. The potential gains of this method are unclear. Moreover, when the observation sites are regularly distributed in space,[2] there is no difference with the standard method.

From now on, the number of local blocks is written N_b. Since the local importance weight vectors $\mathbf{w}_{N_\mathrm{x}:1}$ are constant over each local block, we only need to provide one importance weight per local block and per particle: $w_b(i), (b, i) \in (N_\mathrm{b} : 1) \times (N_\mathrm{e} : 1)$.

4.2.2.3 The local domains

The general idea of DL in the EnKF, as presented in paragraph 2.5.4.1, is that the analysis ensemble for the n-th state variable $\mathbf{E}^\mathsf{a}_n$ is computed using only the observations whose site is located within the n-th local domain. For instance, in two dimensions, a common choice is to define the n-th local domain as a disk, centred at the n-th grid point and whose radius ℓ is a parameter called the **localisation radius**. The same principle can be applied to the LPF–X algorithms: the local domain of the b-th local block is defined as the disk whose centre coincides with that of the b-th block and whose radius ℓ is the localisation radius, a parameter to be determined.

Increasing ℓ means taking more observations into account in the local updates, hence reducing the bias in the analysis step of the LPF–X algorithms. It can also reduce the spatial inhomogeneity by making the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$ smoother in space.

The smoothness of the $\mathbf{w}_{N_\mathrm{b}:1}$ is an important property. Indeed, spatial discontinuities in the $\mathbf{w}_{N_\mathrm{b}:1}$ can lead to spatial discontinuities in the posterior ensemble $\mathbf{E}^\mathsf{a}$ and hence to imbalance. Again lifting ideas from the local EnKF methods, the smoothness of the $\mathbf{w}_{N_\mathrm{b}:1}$ can be improved by tapering the precision of the observations as follows. For the (global) PF, assuming that the observations are independent, the importance weight vector $\mathbf{w}$ is computed

[2] Which is the case in the numerical experiments of chapter 5.

as

$$\mathbf{w} = \pi^{\mathrm{o}}\big(\mathbf{y}\big|\mathbf{E}^{\mathrm{f}}\big) = \prod_{q=1}^{N_{\mathrm{y}}} \pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{E}^{\mathrm{f}}\big), \tag{4.21}$$

where π_q^{o} is the q-th marginal of the observation density π^{o}. Following Poterjoy (2016), for an LPF algorithm, the local importance weight vector $\mathbf{w}_b$ of the b-th local block can be defined as

$$\mathbf{w}_b \triangleq \prod_{q=1}^{N_{\mathrm{y}}} \left[\alpha_q + G\left(\frac{2d_{q,b}}{\ell}\right)\left[\pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{E}^{\mathrm{f}}\big) - \alpha_q\right]\right], \tag{4.22}$$

where α_q is a constant, $d_{q,b}$ is the distance between the q-th site and the centre of the b-th local block, ℓ is the localisation radius, and G is the GC function[3] defined in subsection 2.5.3. Using equation (4.22), we immediately see that if $d_{q,b}$ is larger than ℓ, then $G(2d_{q,b}/\ell) = 0$ and y_q does not contribute to $\mathbf{w}_b$, which is exactly the desired property.

The choice of the constants $\alpha_{N_{\mathrm{y}}:1}$ is delicate. Indeed, for any $q \in (N_{\mathrm{y}}:1)$, α_q affects the transition between $d_{q,b} \to 0$ (full contribution of y_q to $\mathbf{w}_b$) and $d_{q,b} \to \infty$ (no contribution of y_q to $\mathbf{w}_b$). The choice of Poterjoy (2016) is to use $\alpha_q = 1$. However, the more precise the q-th observation y_q, the higher the maximum of $\pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{x}\big)$, which is why α_q should have the same order as the maximum of $\pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{x}\big)$ as suggested by Farchi and Bocquet (2018). Another solution, proposed by Poterjoy et al. (2019), is to define α_q as the mean of $\pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{x}\big)$, which can be estimated as

$$\alpha_q = \frac{1}{N_{\mathrm{e}}}\sum_{i=1}^{N_{\mathrm{e}}} \pi_q^{\mathrm{o}}\big(y_q\big|\mathbf{x}^{\mathrm{f}}(i)\big). \tag{4.23}$$

If the observation error $\boldsymbol{e}^{\mathrm{o}}$ follows a centred Gaussian distribution, with an observation error covariance matrix $\mathbf{R}$, the i-th element of the local importance weight vector $\mathbf{w}_b$ of the b-th local block can alternatively be defined as

$$\ln w_b(i) \triangleq -\frac{1}{2}\left[\mathbf{y} - \mathcal{H}\big(\mathbf{x}^{\mathrm{f}}(i)\big)\right]^{\mathsf{T}} \mathbf{R}_b^{-1}\left[\mathbf{y} - \mathcal{H}\big(\mathbf{x}^{\mathrm{f}}(i)\big)\right], \tag{4.24}$$

where the tapered observation observation error covariance matrix $\mathbf{R}_b$ is defined in the same way as for the implementation of DL in the EnKF, as described in paragraph 2.5.4.1. When the observations are independent, $\mathbf{R}$ is diagonal, and equation (4.24) simplifies into

$$\ln w_b(i) = -\sum_{q=1}^{N_{\mathrm{y}}} G\left(\frac{2d_{q,b}}{\ell}\right)\frac{\left[y_q - \mathcal{H}_q\big(\mathbf{x}^{\mathrm{f}}(i)\big)\right]^2}{2r_q^2}, \tag{4.25}$$

where r_q^2 and $\mathcal{H}_q$ are the variance and the observation operator corresponding to the q-th observation. This formula is introduced by Shen et al. (2017) as a new formulation of the $\mathbf{w}_{N_{\mathrm{b}}:1}$, however, one should keep in mind that this formula is directly inspired from the application of DL to the EnKF, for example with the LETKF algorithm.

Both equation (4.22) and equation (4.24) have advantages and drawbacks. The *generic*

[3] Here, and everywhere else in this chapter, the GC function G could be replaced by any other smooth taper function.

Algorithm 4.1: Analysis step for a generic LPF–X algorithm.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\circ}$, $\ell$, block shape
for $b = 1$ to $N_{\mathrm{b}}$ do
    $\mathbf{w}_b \leftarrow$ equation (4.22) or (4.24)
    $\mathbf{E}^{\mathsf{r}}_{|b} \leftarrow \texttt{Resampling}\big(\mathbf{w}_b, \mathbf{E}^{\mathsf{f}}_{|b}\big)$
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \texttt{Assembling}\big(\mathbf{E}^{\mathsf{r}}_{|N_{\mathrm{b}}:1}\big)$
Output: $\mathbf{E}^{\mathsf{a}}$
```

formulation, equation (4.22), is defined if the observation error $\boldsymbol{e}^{\circ}$ is not Gaussian, but only if the observations are independent. By contrast, in the *Gaussian* formulation, equation (4.24), the observations can be correlated, but the $\boldsymbol{e}^{\circ}$ must be Gaussian. If the observation error $\boldsymbol{e}^{\circ}$ is Gaussian and if the observations are independent, both formulations are equivalent in the following cases:

- in the limit of a zero localisation radius, $\ell \to 0$, and an infinite observation variance, $\forall q \in (N_{\mathrm{y}} : 1)$, $r_q^2 \to \infty$;
- in the limit of an infinite localisation radius $\ell \to \infty$.

In other cases, they may lead to different performances of the resulting LPF–X algorithms.

Remark 20. The terminology adopted in this paragraph (disk, radius, . . .) fits two-dimensional spatial spaces. Yet most geophysical models have a three-dimensional spatial structure, with typical uneven vertical scales that are usually much shorter than horizontal scales. For these models, the geometry of the local domains should be adapted accordingly.

4.2.2.4 The generic LPF–X algorithm

Algorithm 4.1 describes the analysis step for a generic LPF–X algorithm, in which the matrices $\mathbf{E}^{\mathsf{f}}_{|b}$ and $\mathbf{E}^{\mathsf{a}}_{|b}$ designate the restriction of $\mathbf{E}^{\mathsf{f}}$ and $\mathbf{E}^{\mathsf{a}}$ to the b-th local block.[4] The definition of local blocks and domains is illustrated, in a two-dimensional physical space, in figure 4.3.

4.2.2.5 Beating the curse of dimensionality

The feasibility of LPF–X algorithms is discussed by Rebeschini and van Handel (2015) through the example of their block PF algorithm. In this algorithm, the distinction between local

[4] In other words, the rows of $\mathbf{E}^{\mathsf{f}}_{|b}$ and $\mathbf{E}^{\mathsf{a}}_{|b}$ are the rows of $\mathbf{E}^{\mathsf{f}}$ and $\mathbf{E}^{\mathsf{a}}$ corresponding to the grid points located with the b-th local block.

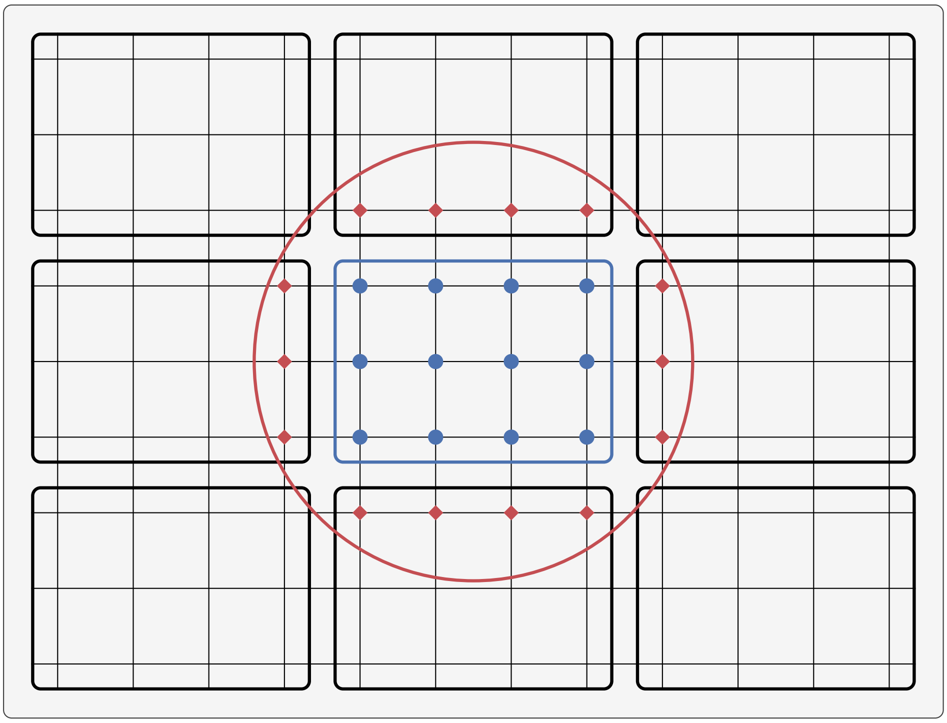

Figure 4.3: Illustration of the definition of the geometry for LPF–X algorithms in a two-dimensional physical space. The local blocks are shown with black rectangles. The focus is on the local block in the middle (blue rectangle), which gathers 12 grid points (blue circles). The corresponding local domain is circumscribed by a red circle, with potential observation sites outside the block (red diamonds).

blocks and domains does not exist. The GC function G is replaced by a top-hat function, and the resampling step is performed independently for each block, regardless of the boundaries between blocks.

As presented in section 3.1, the mean squared error on the IS estimates is the sum of the bias term and the variance term. The main mathematical result is that, under minimal conditions, the bias term is related to the block boundaries and decreases exponentially with the diameter of the blocks (measured in number of grid points). It is due to the fact that the analysis does not exactly follows Bayes' theorem any more, because only a subset of observations is used to update each block. The exponential decrease is a demonstration of the *decay of correlations* property. The variance term is computed using the central limit theorem, and scales with $\exp K/N_\mathrm{e}$. In the global PF, K is related the dimension of state space N_x, whereas here K is the number of grid points inside each local block. This implies that LPF–X algorithms can indeed beat the curse of dimensionality with a reasonably large number of particles N_e.

4.2.3 The local resampling

As mentioned in subsection 4.2.1, resampling from the distribution whose pdf is given by equation (4.20) does not cause any theoretical or technical issue. Indeed, the resampling step is performed independently for each local block, and the (global) posterior ensemble $\mathbf{E}^\mathsf{a}$ is obtained by assembling the locally resampled ensembles $\mathbf{E}^\mathsf{r}_{|N_\mathrm{b}:1}$. This is the strategy described in algorithm 4.1.

By doing so, adjacent local blocks are uncoupled. This is beneficial, because uncoupling is a way to counteract the curse of dimensionality. On the other hand, regardless of the spatial smoothness of the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$, blind assembling is likely to yield unphysical discontinuities, and hence imbalance, in $\mathbf{E}^\mathsf{a}$. More precisely, while assembling the $\mathbf{E}^\mathsf{r}_{|N_\mathrm{b}:1}$, there is a high probability of obtaining **composite** particles. A posterior particle is said to be a composite particle if it is composed of the i-th prior particle $\mathbf{x}^\mathsf{f}(i)$ on one local block, and of the j-th prior particle $\mathbf{x}^\mathsf{f}(j)$ on an other local block, with $j \neq i$. In that case, there is no guarantee that $\mathbf{x}^\mathsf{f}(i)$ and $\mathbf{x}^\mathsf{f}(j)$ are *close*, and that assembling them will represent a physical state. A pathological example is illustrated, in one dimension, in figure 4.4.

In order to mitigate the unphysical discontinuities – and hence to mitigate the imbalance – the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$ must be spatially smooth, as already mentioned in subsection 4.2.2. Furthermore, the resampling method must have some *regularity*, in order to preserve part of the spatial structure held in the prior ensemble $\mathbf{E}^\mathsf{f}$. Potential solutions are presented hereafter.

4.2.3.1 Applying a smoothing-by-weights step

Following the idea of Penny and Miyoshi (2016), a first solution is to add a **smoothing-by-weights** after the resampling step. The goal of this additional step is to smooth out the potential unphysical discontinuities by averaging in space the locally resampled ensembles $\mathbf{E}^\mathsf{r}_{|N_\mathrm{b}:1}$.

Let $\mathbf{E}^\mathsf{r}_b$ be the $N_\mathrm{x} \times N_\mathrm{e}$ matrix obtained by applying the resampling method to the (global) prior ensemble $\mathbf{E}^\mathsf{f}$, weighted by the local importance weight vector $\mathbf{w}_b$ of the b-th local block.

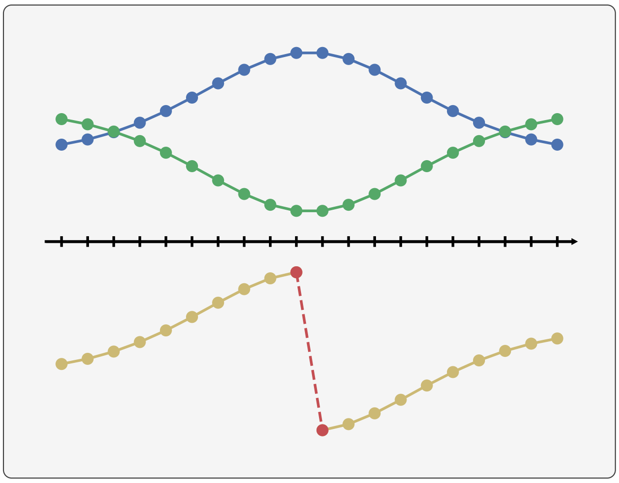

Figure 4.4: Illustration of the formation of a composite particle in one dimension. The yellow particle is the concatenation of the blue particle (left part) and of the green particle (right part). In this situation, a large unphysical discontinuity appears at the boundary (red dashed line).

The matrix $\mathbf{E}_b^{\mathsf{r}}$ is different from the matrix $\mathbf{E}_{|b}^{\mathsf{r}}$ introduced in subsection 4.2.2: indeed, $\mathbf{E}_{|b}^{\mathsf{r}}$ is the restriction of $\mathbf{E}_b^{\mathsf{r}}$ to the b-th local block. The smoothed ensemble $\mathbf{E}^{\mathsf{s}}$ is then defined as the $N_{\mathrm{x}} \times N_{\mathrm{e}}$ matrix whose n-th row, i-th column element is

$$[\mathbf{E}^{\mathsf{s}}]_{n,i} \triangleq \frac{\displaystyle\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right)[\mathbf{E}_b^{\mathsf{r}}]_{n,i}}{\displaystyle\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right)}, \tag{4.26}$$

where $d_{n,b}$ is the physical distance between the n-th grid point and the centre of the b-th local block, and where ℓ^{s} is the **smoothing radius**, a parameter potentially different from the localisation radius ℓ. Using this definition, the posterior ensemble $\mathbf{E}^{\mathsf{a}}$ is computed as

$$\mathbf{E}^{\mathsf{a}} = \alpha^{\mathsf{s}}\mathbf{E}^{\mathsf{s}} + (1-\alpha^{\mathsf{s}})\mathbf{E}^{\mathsf{r}}, \tag{4.27}$$

where $\alpha^{\mathsf{s}} \in [0,1]$ is the **smoothing strength**, a parameter to be determined, and where the resampled ensemble $\mathbf{E}^{\mathsf{r}}$ is the ensemble obtained by assembling the locally resampled ensembles $\mathbf{E}_{|N_{\mathrm{b}}:1}^{\mathsf{r}}$.

When $\alpha^{\mathsf{s}} = 0$, no smoothing is performed and the update described in the generic LPF–X algorithm is recovered. When $\alpha^{\mathsf{s}} = 1$, $\mathbf{E}^{\mathsf{r}}$ is totally replaced by $\mathbf{E}^{\mathsf{s}}$. Therefore, the smoothing strength controls the intensity of the smoothing. Algorithm 4.2 describes the analysis step for a generic LPF–X algorithm with smoothing-by-weights. The original LPF algorithm by Penny and Miyoshi (2016) can be recovered if the following conditions are satisfied:

- the local blocks have a size of 1 grid point;

Algorithm 4.2: Analysis step for a generic LPF–X algorithm with smoothing-by-weights.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\mathsf{o}}$, $\ell$, block shape, $\ell^{\mathsf{s}}$, $\alpha^{\mathsf{s}}$
for $b = 1$ to $N_{\mathrm{b}}$ do
    $\mathbf{w}_b \leftarrow$ equation (4.22) or (4.24)
    $\mathbf{E}^{\mathsf{r}}_{|b} \leftarrow \texttt{Resampling}\big(\mathbf{w}_b, \mathbf{E}^{\mathsf{f}}_{|b}\big)$
    $\mathbf{E}^{\mathsf{r}}_{b} \leftarrow \texttt{Resampling}\big(\mathbf{w}_b, \mathbf{E}^{\mathsf{f}}_{b}\big)$
end
$\mathbf{E}^{\mathsf{r}} \leftarrow \texttt{Assembling}\big(\mathbf{E}^{\mathsf{r}}_{|N_{\mathrm{b}}:1}\big)$
$\mathbf{E}^{\mathsf{s}} \leftarrow$ equation (4.26)
$\mathbf{E}^{\mathsf{a}} \leftarrow \alpha^{\mathsf{s}}\mathbf{E}^{\mathsf{s}} + (1 - \alpha^{\mathsf{s}})\mathbf{E}^{\mathsf{r}}$
Output: $\mathbf{E}^{\mathsf{a}}$
```

- the local importance weight vectors $\mathbf{w}_{N_{\mathrm{b}}:1}$ are computed using the Gaussian formulation, equation (4.24);
- the GC function G is replaced by a top-hat function in equation (4.24) and in equation (4.26);
- the resampling method is the systematic resampling algorithm (algorithm 3.4);
- the smoothing radius ℓ^{s} is set to be equal to the localisation radius ℓ;
- the smoothing strength α^{s} is set to 1/2.

Algorithm 4.2 is a generalisation of their original LPF algorithm. The smoothing-by-weights step is an ad hoc fix to reduce potential unphysical discontinuities after they have been introduced in the local resampling step. Its necessity hints that there is room for improvement in the design of the local resampling methods.

4.2.3.2 Simplification and generalisation of the smoothing-by-weights step

It turns out that the smoothing-by-weights step described in the previous paragraph can be simplified as follows. Let ψ_b be the resampling map for the b-th local block, that is the map (computed with $\mathbf{w}_b$) such that $\psi_b(i)$ is the index of the i-th selected particle in the resampling of the b-th local block. The construction of the resampling map ψ is explained in algorithms 3.3 and 3.4 when using the multinomial or the systematic resampling algorithm.

With this notation, the n-th element of the i-th particle in the smoothed ensemble $\mathbf{E}^{\mathsf{s}}$ can be computed as

$$x_n^{\mathsf{s}}(i) = \frac{\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right) x_n^{\mathsf{f}}\big(\psi_b(i)\big)}{\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right)}, \tag{4.28}$$

and therefore the n-th element of the i-th particle in the posterior ensemble $\mathbf{E}^{\mathsf{a}}$ is given by

$$x_n^{\mathsf{a}}(i) = (1-\alpha^{\mathsf{s}})\, x_n^{\mathsf{f}}\big(\psi_n(i)\big) + \alpha^{\mathsf{s}} \frac{\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right) x_n^{\mathsf{f}}\big(\psi_b(i)\big)}{\sum_{b=1}^{N_{\mathrm{b}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{s}}}\right)}, \tag{4.29}$$

where ψ_n is defined as the resampling map corresponding to the local block in which the n-th variable is located. This expression can be used to efficiently implement algorithm 4.2.

Furthermore, the smoothing-by-weights step is non-invasive, meaning that it is added *after* the resampling step. As a consequence, it can be straightforwardly generalised to the case where the resampling step is replaced by a linear transformation or transport step, as presented in paragraphs 4.2.3.4 and 4.2.3.5.

4.2.3.3 Refinements of the resampling methods

In this paragraph, we examine several properties of the resampling methods which might help dealing with the discontinuity issue.

- A resampling algorithm is said to be **balanced** if, for all $i \in (N_{\mathrm{e}}\!:\!1)$, the number of offspring of the i-th prior particle $\mathbf{x}^{\mathsf{f}}(i)$ does not differ by more than one unity from $N_{\mathrm{e}}\bar{w}(i)$. For example, this is the case of the systematic resampling algorithm (algorithm 3.4), but not of the multinomial resampling algorithm (algorithm 3.3).
- A resampling algorithm is said to be **adjustment-minimising** if the indices of the resampled particles $\mathbf{x}^{\mathsf{r}}(N_{\mathrm{e}}\!:\!1)$ are reordered to maximise the number of indices $i \in (N_{\mathrm{e}}\!:\!1)$ such that the i-th resampled particle $\mathbf{x}^{\mathsf{r}}(i)$ is a copy of the i-th prior particle $\mathbf{x}^{\mathsf{f}}(i)$. Both the multinomial and the systematic resampling algorithm can be simply modified to yield adjustment-minimising resampling algorithms.
- While performing the resampling step independently for each local block, one can use the same random number(s) in the local resampling of each local block.

Using the same random number(s) for the local resampling of all local blocks avoids a stochastic source of unphysical discontinuity. Choosing balanced and adjustment-minimising resampling algorithms is an attempt to include some kind of continuity in the map

$$\{\text{local weights}\} \mapsto \{\text{locally resampled particles}\}, \tag{4.30}$$

by minimising the occurrences of composite particles. However, these properties cannot eliminate all sources of unphysical discontinuity. Indeed, ultimately, composite particles will be built (if not, then localisation would not be necessary) and there is no mechanism to reduce the unphysical discontinuities in them. These properties have been first introduced in the *naive* local ensemble Kalman particle filter (EnKPF) by S. Robert and Künsch (2017).

4.2.3.4 Using a linear transformation step instead of the resampling step

In this paragraph, we study the benefits of replacing the local resampling step by a linear transformation step, using the example of the optimal ensemble coupling, described in paragraph 3.3.3.1.

Applying the optimal ensemble coupling for the local update of the b-th local block results in an optimisation problem which consists in finding the LET matrix $\mathbf{T}_{\mathrm{e}}^{*} \in \mathbb{T}(\mathbf{w}_b)$ minimising the cost function

$$\mathscr{J}_b\big[\mathbf{E}^{\mathsf{f}}\big]\big(\mathbf{T}_{\mathrm{e}}\big) \triangleq \sum_{i=1}^{N_{\mathrm{e}}}\sum_{j=1}^{N_{\mathrm{e}}}\big[\mathbf{T}_{\mathrm{e}}\big]_{i,j}\big[\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big)\big]_{i,j}. \tag{4.31}$$

In the (global) ETPF algorithm, the cost coefficients $\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big)$ are defined as the squared $\mathcal{L}^2$-distance between the particles in the prior ensemble $\mathbf{E}^{\mathsf{f}}$. Since the update here is local, it seems more appropriate to use a local cost coefficient, such as

$$\forall (i,j) \in (N_{\mathrm{e}} : 1)^2, \quad \big[\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big)\big]_{i,j} \triangleq \sum_{n=1}^{N_{\mathrm{x}}} G\left(\frac{d_{n,b}}{\ell^{\mathsf{d}}}\right)\Big[x_n^{\mathsf{f}}(i) - x_n^{\mathsf{f}}(j)\Big]^2, \tag{4.32}$$

where $d_{n,b}$ is the distance between the n-th grid point and the centre of the b-th block, and ℓ^{d} is the **distance radius**, a parameter to be determined.

Algorithm 4.3 describes the analysis step for a generic LPF–X algorithms using optimal ensemble coupling. As a result from the minimisation, on each local block $b \in (N_{\mathrm{b}} : 1)$, $\mathbf{T}_{\mathrm{e}}^{*}$ establishes a strong and deterministic connection between $\mathbf{E}_{|b}^{\mathsf{f}}$ and $\mathbf{E}_{|b}^{\mathsf{r}}$, the local prior and posterior ensembles. Therefore, in this algorithm the spatial coherence is, at least partially, transferred from the prior ensemble $\mathbf{E}^{\mathsf{f}}$ to the posterior ensemble $\mathbf{E}^{\mathsf{a}}$.

Remark 21. Localisation has been first included in the ETPF algorithm by Cheng and Reich (2015), in a similar way as the block localisation formalism. Hence Algorithm 4.3 can be seen as a generalisation of the local ETPF of Cheng and Reich (2015) which includes the concept of local blocks.

4.2.3.5 Using a transport step instead of the resampling step

In this paragraph, we study the benefits of replacing the local resampling step by a transport step, as described in paragraph 3.3.3.2. In more than one dimension, the transport problem, formulated by equation (3.64), is highly non-trivial, which is why in this paragraph, we choose to restrict ourselves to the one-dimensional case. Therefore, local blocks cannot gather more than one state variable and hence there is no distinction between local bocks and state variables.

Algorithm 4.3: Analysis step for a generic LPF–X algorithm using optimal ensemble coupling.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\mathsf{o}}$, $\ell$, block shape, $\ell^{\mathsf{d}}$
for $b = 1$ to $N_{\mathrm{b}}$ do
    $\mathbf{w}_b \leftarrow$ equation (4.22) or (4.24)
    $\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big) \leftarrow$ equation (4.32)
    $\mathbf{T}_{\mathrm{e}}^{*} \leftarrow \underset{\mathbf{T}_{\mathrm{e}}\in\mathbb{T}(\mathbf{w}_b)}{\arg\min}\, \mathcal{J}_b\big[\mathbf{E}^{\mathsf{f}}\big]\big(\mathbf{T}_{\mathrm{e}}\big)$
    $\mathbf{E}^{\mathsf{r}}_{|b} \leftarrow \mathbf{E}^{\mathsf{f}}_{|b}\mathbf{T}_{\mathrm{e}}$
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \mathtt{Assembling}\big(\mathbf{E}^{\mathsf{r}}_{|N_{\mathrm{b}}:1}\big)$
Output: $\mathbf{E}^{\mathsf{a}}$
```

In this case, the update for the n-th variable is performed with the transport map $\mathcal{T}_n$ such that

$$\overline{\pi}^{\mathsf{a}}_n = \mathcal{T}_n^{\#}\overline{\pi}^{\mathsf{f}}_n. \tag{4.33}$$

In this equation, $\overline{\pi}^{\mathsf{a}}_n$ is the n-th marginal of the empirical analysis density, as introduced in subsection 4.2.1, and $\overline{\pi}^{\mathsf{f}}_n$ is the n-th marginal of the empirical forecast density, which, by analogy with $\overline{\pi}^{\mathsf{a}}_n$, must be defined as

$$\overline{\pi}^{\mathsf{f}}_n(x_n) = \sum_{i=1}^{N_{\mathrm{e}}} \delta\big(x_n - x^{\mathsf{f}}_n(i)\big). \tag{4.34}$$

In order to obtain a non-discrete $\mathcal{T}_n$, continuous representations are needed for $\overline{\pi}^{\mathsf{f}}_n$ and $\overline{\pi}^{\mathsf{a}}_n$. An appealing approach is to use the regularisation framework presented in paragraph 3.3.2.2 to define $\overline{\pi}^{\mathsf{f}}_n$ and $\overline{\pi}^{\mathsf{a}}_n$ as

$$\overline{\pi}^{\mathsf{f}}_n(x_n) \triangleq \sum_{i=1}^{N_{\mathrm{e}}} \mathcal{K}\left[\frac{x_n - x^{\mathsf{f}}_n(i)}{h^{\mathsf{f}}\sigma^{\mathsf{f}}_n}\right], \tag{4.35}$$

$$\overline{\pi}^{\mathsf{a}}_n(x_n|\mathbf{y}) \triangleq \sum_{i=1}^{N_{\mathrm{e}}} \overline{w}_n(i)\,\mathcal{K}\left[\frac{x_n - x^{\mathsf{f}}_n(i)}{h^{\mathsf{a}}\sigma^{\mathsf{a}}_n}\right]. \tag{4.36}$$

The map $\mathcal{K}$ is the regularisation kernel, to be defined, h^{f} and h^{a} are the (normalised) forecast and analysis regularisation **bandwidths**, two parameters to be determined, and σ^{f}_n and σ^{a}_n are the empirical standard deviations of the ensembles $\big\{x^{\mathsf{f}}_n(i), i \in (N_{\mathrm{e}}:1)\big\}$ and

$\{(w_n(i), x^{\mathsf{f}}_n(i)), i \in (N_{\mathrm{e}} : 1)\}$, given by

$$\left(\sigma^{\mathsf{f}}_n\right)^2 \triangleq \frac{1}{N_{\mathrm{e}} - 1} \sum_{i=1}^{N_{\mathrm{e}}} \left[x^{\mathsf{f}}_n(i) - \frac{1}{N_{\mathrm{e}}} \sum_{j=1}^{N_{\mathrm{e}}} x^{\mathsf{f}}_n(j) \right]^2, \tag{4.37}$$

$$\left(\sigma^{\mathsf{a}}_n\right)^2 \triangleq \frac{1}{1 - \bar{\mathbf{w}}_n^{\mathsf{T}} \bar{\mathbf{w}}_n} \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}_n(i) \left[x^{\mathsf{f}}_n(i) - \sum_{j=1}^{N_{\mathrm{e}}} \bar{w}_n(j)\, x^{\mathsf{f}}_n(j) \right]^2. \tag{4.38}$$

As mentioned in paragraph 3.3.3.2, the transport condition, equation (4.33) may admit more than one solution. Therefore, we choose to use, for the update for the n-th variable, the transport application $\mathcal{T}^*_n$ which minimises the cost function

$$\mathcal{J}_n(\mathcal{T}_n) = \int \left(x_n - \mathcal{T}_n(x_n)\right)^2 \bar{\pi}^{\mathsf{f}}_n(x_n)\, \mathrm{d}x_n, \tag{4.39}$$

over all transport applications $\mathcal{T}_n$ satisfying equation (4.33). In the statistical literature, the optimal transport application $\mathcal{T}^*_n$ is also known as the **anamorphosis** between $\bar{\pi}^{\mathsf{f}}_n$ and $\bar{\pi}^{\mathsf{a}}_n$, and it can be computed as

$$\mathcal{T}^*_n = \left(\phi^{\mathsf{a}}_n\right)^{-1} \circ \phi^{\mathsf{f}}_n, \tag{4.40}$$

where ϕ^{f}_n and ϕ^{f}_n are the cumulative density function (cdf)s of $\bar{\pi}^{\mathsf{f}}_n$ and $\bar{\pi}^{\mathsf{a}}_n$.

According to the KDE theory (Silverman 1986; Musso et al. 2001), when the underlying distribution is Gaussian, the optimal shape for the regularisation kernel $\mathcal{K}$ is the Epanechnikov kernel (quadratic functions). Yet there is no reason to think that this will also be the case for $\bar{\pi}^{\mathsf{f}}_n$ and $\bar{\pi}^{\mathsf{a}}_n$. Besides, the Epanechnikov kernel, having a finite support, generally leads to a poor representation of the distribution tails, and it is a potential source of indetermination in the definition of the cdfs. That is why a more common approach is to use a Gaussian regularisation kernel $\mathcal{K}$. However, in this case, the cost of computing the cdf of $\mathcal{K}$ (namely, the error function) becomes significant. Hence, as an alternative, we choose to use the Student's t-distribution with two degrees of freedom. Its pdf is visually similar to a Gaussian density with heavy tails, as illustrated in figure 4.5, and its cdf is fast to compute. Moreover, it was shown to yield a better representation of the forecast density π^{f} than a Gaussian distribution in an EnKF context (Bocquet et al. 2015). The use of a regularisation kernel $\mathcal{K}$ different from the Dirac kernel δ is necessary to obtain continuous transport maps $\mathcal{T}_{N_{\mathrm{x}}:1}$, although it introduces an additional bias in the analysis. The magnitude of the regularisation is controlled by the forecast and analysis regularisation bandwidths h^{f} and h^{a}. Dirac kernels are recover in the limit $h^{\mathsf{f}} \to 0$ and $h^{\mathsf{a}} \to 0$.

Algorithm 4.4 describes the resulting analysis step for a generic LPF–X algorithm using anamorphosis. The anamorphosis is, as well as the optimal ensemble coupling presented in the previous paragraph, a deterministic transformation. This means that unphysical discontinuities due to different random realisations over the grid points are avoided. As explained by Poterjoy (2016), for any state variable $n \in (N_{\mathrm{x}} : 1)$, the posterior particles $x^{\mathsf{a}}_n(N_{\mathrm{e}} : 1)$ obtained with the anamorphosis have the same quantiles as the prior particles $x^{\mathsf{f}}_n(N_{\mathrm{e}} : 1)$. The quantile property should, to some extent, be regular in space – for example if the spatial discretisation is fine enough – and this kind of regularity is transferred from the

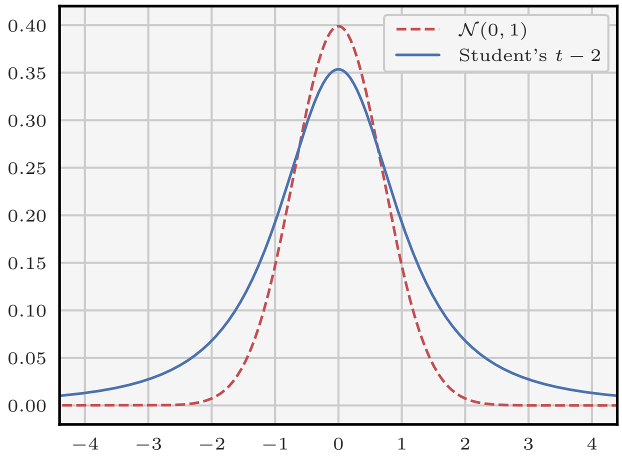

Figure 4.5: Illustration of the pdfs of the Student's t-distribution with two degrees of freedom (in blue), and of the Gaussian distribution $\mathcal{N}[0,1]$ (in red). Visually, both pdfs are similar, but the pdf of the Student's t-distribution with two degrees of freedom has heavier tails.

prior ensemble $\mathbf{E}^{\mathsf{f}}$ to the posterior ensemble $\mathbf{E}^{\mathsf{a}}$.

The refinements of the resampling methods, introduced in paragraph 4.2.3.3, are designed to minimise the number of unphysical discontinuities in the local resampling step. The goal of the additional smoothing-by-weights step, introduced in paragraph 4.2.3.1, is to mitigate potential unphysical discontinuities after they have been introduced. By contrast, the local transformation methods based on optimal transport – both the optimal ensemble coupling introduced in the previous paragraph and the anamorphosis introduced in this paragraph – are designed to mitigate the unphysical discontinuities themselves. This theoretical advantage is largely validated by the numerical experiments of chapter 5.

Remark 22. The design of the local transformation based on anamorphosis is inspired from the kernel density distribution mapping (KDDM) step of the LPF algorithm of Poterjoy (2016), which is introduced in subsection 4.3.2. However, the use of optimal transport has different purposes. As presented here, the anamorphosis transformation is used to transport the prior ensemble $\mathbf{E}^{\mathsf{f}}$ towards the empirical analysis density $\bar{\pi}^{\mathsf{a}}$, whereas the KDDM step described by Poterjoy (2016) is designed to correct the posterior ensemble $\mathbf{E}^{\mathsf{a}}$ (which has already been transformed) with consistent high-order statistical moments.

4.3 The LPF–Y algorithms: a sequential localisation framework

In the block localisation framework introduced in the previous section, the ensemble update is performed independently for each local block. In this section, we adopt a different approach: after the sampling step, the observations are assimilated sequentially, with the constraint that each observation should only influence the neighbourhood grid points. The general idea

Algorithm 4.4: Analysis step for a generic LPF–X algorithm using anamorphosis.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\mathsf{o}}$, $\ell$, $h^{\mathsf{f}}$, $h^{\mathsf{a}}$
for $n = 1$ to $N_{\mathrm{x}}$ do
    $\mathbf{w}_n \leftarrow$ equation (4.22) or (4.24)
    $\sigma_n^{\mathsf{f}} \leftarrow$ equation (4.37)
    $\sigma_n^{\mathsf{a}} \leftarrow$ equation (4.38)
    for $i = 1$ to $N_{\mathrm{e}}$ do
        $x_n^{\mathsf{r}}(i) \leftarrow \big(\phi_n^{\mathsf{a}}\big)^{-1} \circ \phi_n^{\mathsf{f}}\big(x_n^{\mathsf{f}}(i)\big)$
    end
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \mathtt{Assembling}\big(x_{N_{\mathrm{x}}:1}^{\mathsf{r}}(N_{\mathrm{e}}:1)\big)$
Output: $\mathbf{E}^{\mathsf{a}}$
```

is that this sequential framework can be used to design local updates as smooth as possible in space, which would mitigate the imbalance.

Following the starting point for LPF–X algorithms, the sampling step is performed with an equally weighted ensemble $\mathbf{E}$, using the standard proposal. Again, this means that the sampling step is equivalent to a forecast step, and that the local sequential updates can be considered as the analysis step. Such LPF algorithms are called LPF–Y algorithms, where the –Y extension emphasises the fact that there is one update per observation.

In this section, we keep the simplifications in notation introduced in the previous section. The new localisation framework is introduced in subsection 4.3.1, and two examples of LPF–Y algorithms are provided in subsections 4.3.2 and 4.3.3.

4.3.1 The sequential localisation framework

4.3.1.1 Partitioning the state space

Following S. Robert and Künsch (2017), for the assimilation of the q-th observation y_q, the state space $\mathbb{R}^{N_{\mathrm{x}}}$ is divided into three regions.

1. The first region U contains all variables with a direct contribution to y_q. If the observation operator $\mathcal{H}$ is linear, with matrix $\mathbf{H}$, this corresponds to the columns of $\mathbf{H}$ with non-zero elements on the q-th row.

2. The second region V gathers all variables which are deemed correlated to those in U.

3. The third region W contains all remaining variables.

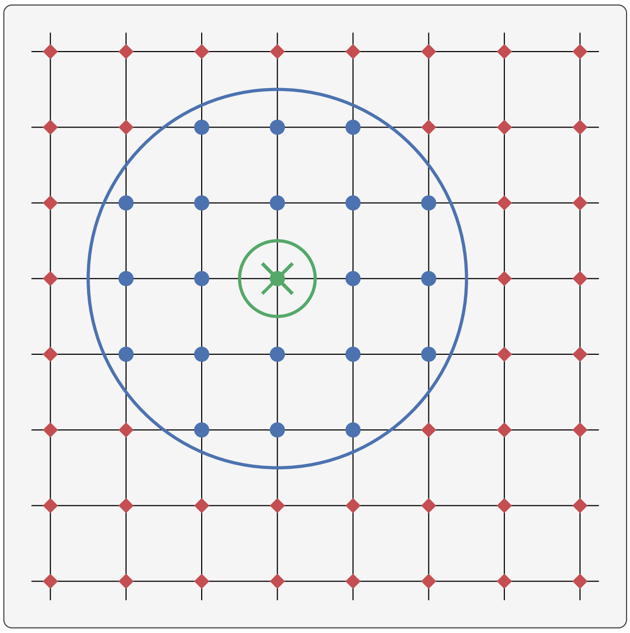

Figure 4.6: Illustration of the q-th UVW partition for LPF–Y algorithms in a two-dimensional physical space. The site of the q-th observation y_q is depicted by a green mark. The local regions U and V are circumscribed by the thick green and blue circles, and contain 1 and 20 grid points (green and blue circles), respectively. The global region W contain the remaining 43 grid points (red diamonds).

The meaning of *correlated* is to be understood as a prior hypothesis, where we define a valid localisation matrix $\boldsymbol{\rho}$ representing the decay of correlations in the physical space. Most of the time, $\boldsymbol{\rho}$ is constructed exactly as for CL in the EnKF, using equation (2.54), in which the localisation radius ℓ is a free parameter.

The UVW partition of the state space $\mathbb{R}^{N_\mathsf{x}}$ is a generalisation of the original LG partition introduced by Bengtsson et al. (2003), in which the regions U and V are gathered into a single region L, the local region of y_q, and the region W is called G, the global region. Figure 4.6 illustrates this UVW partition in a two-dimensional physical space. We emphasise that both the LG and the UVW partitions of the state space $\mathbb{R}^{N_\mathsf{x}}$ depend on y_q, and therefore they are fundamentally different from the local state block framework introduced in subsection 4.2.2.

In the sequential framework, the goal of the q-th update is to estimate the q-th conditional density $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$, in which the conditioning with respect to the previous observations $\boldsymbol{y}_{q-1:1}$[5] is implicit. Therefore, in the following paragraphs, we study the factorisation of $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$. For simplicity, the restriction of a vector $\mathbf{x} \in \mathbb{R}^{N_\mathsf{x}}$ to a region A of the state space, $\mathbf{x}_{n\in\mathsf{A}}$, is written $\mathbf{x}_\mathsf{A}$.

[5] Which is an empty set if $q = 1$.

Algorithm 4.5: Analysis step for a generic LPF–Y algorithm using the LG partition.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\circ}$, $\boldsymbol{\rho}$
$\mathbf{E} \leftarrow \mathbf{E}^{\mathsf{f}}$                        // initialise the sequential updates
for $q = 1$ to $N_{\mathrm{y}}$ do
    From $\boldsymbol{\rho}$, build the LG partition    // as described in paragraph 4.3.1.1
    for $i = 1$ to $N_{\mathrm{e}}$ do
        Do not update $\mathbf{x}_{\mathsf{G}}(i)$
        Update $\mathbf{x}_{\mathsf{L}}(i)$ conditionally to $\mathbf{x}_{\mathsf{G}}(i)$ and $y_q$
    end
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \mathbf{E}$
Output: $\mathbf{E}^{\mathsf{a}}$
```

4.3.1.2 The conditional density with the LG partition

Without loss of generality, the q-th conditional density $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ is decomposed into

$$\pi[\boldsymbol{x}|\boldsymbol{y}_q] = \pi[\boldsymbol{x}_{\mathsf{L}}, \boldsymbol{x}_{\mathsf{G}}|\boldsymbol{y}_q], \tag{4.41}$$

$$= \pi[\boldsymbol{x}_{\mathsf{L}}|\boldsymbol{x}_{\mathsf{G}}, \boldsymbol{y}_q]\, \pi[\boldsymbol{x}_{\mathsf{G}}|\boldsymbol{y}_q]. \tag{4.42}$$

In a localisation context, it seems reasonable to assume that the U region and $\boldsymbol{y}_q$ are independent, which means that

$$\pi[\boldsymbol{x}_{\mathsf{G}}|\boldsymbol{y}_q] = \pi[\boldsymbol{x}_{\mathsf{G}}]. \tag{4.43}$$

The resulting factorisation for $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ is

$$\pi[\boldsymbol{x}|\boldsymbol{y}_q] = \pi[\boldsymbol{x}_{\mathsf{L}}|\boldsymbol{x}_{\mathsf{G}}, \boldsymbol{y}_q]\, \pi[\boldsymbol{x}_{\mathsf{G}}]. \tag{4.44}$$

From this factorisation, we conclude the generic analysis method described in algorithm 4.5, in which the ensemble $\mathbf{E}$ is gradually updated from $\mathbf{E}^{\mathsf{f}}$ to $\mathbf{E}^{\mathsf{a}}$ using N_{y} local sequential updates.

4.3.1.3 The conditional density with the UVW partition

With the UVW partition, the q-th conditional density $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ is factored as

$$\begin{aligned}
\pi[\boldsymbol{x}|\boldsymbol{y}_q] &= \pi[\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{V}}, \boldsymbol{x}_{\mathsf{W}}|\boldsymbol{y}_q], && (4.45)\\
&= \frac{\pi[\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{V}}, \boldsymbol{x}_{\mathsf{W}}, \boldsymbol{y}_q]}{\pi[\boldsymbol{y}_q]}, && (4.46)\\
&= \frac{\pi[\boldsymbol{y}_q|\boldsymbol{x}]\,\pi[\boldsymbol{x}_{\mathsf{V}}|\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]\,\pi[\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]}{\pi[\boldsymbol{y}_q]}, && (4.47)\\
&= \frac{\pi[\boldsymbol{y}_q|\boldsymbol{x}_{\mathsf{U}}]\,\pi[\boldsymbol{x}_{\mathsf{V}}|\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]\,\pi[\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]}{\pi[\boldsymbol{y}_q]}. && (4.48)
\end{aligned}$$

If one assumes that the U and W regions are not only uncorrelated but also independent, then one can make the additional factorisation

$$\pi[\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}] = \pi[\boldsymbol{x}_{\mathsf{U}}]\,\pi[\boldsymbol{x}_{\mathsf{W}}]. \tag{4.49}$$

Finally, $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ is

$$\begin{aligned}
\pi[\boldsymbol{x}|\boldsymbol{y}_q] &= \frac{\pi[\boldsymbol{y}_q|\boldsymbol{x}_{\mathsf{U}}]\,\pi[\boldsymbol{x}_{\mathsf{U}}]}{\pi[\boldsymbol{y}_q]}\pi[\boldsymbol{x}_{\mathsf{V}}|\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]\,\pi[\boldsymbol{x}_{\mathsf{W}}], && (4.50)\\
&= \pi[\boldsymbol{x}_{\mathsf{U}}|\boldsymbol{y}_q]\,\pi[\boldsymbol{x}_{\mathsf{V}}|\boldsymbol{x}_{\mathsf{U}}, \boldsymbol{x}_{\mathsf{W}}]\,\pi[\boldsymbol{x}_{\mathsf{W}}]. && (4.51)
\end{aligned}$$

From this factorisation, we conclude the generic analysis method described in algorithm 4.6, in which, again, the ensemble $\mathbf{E}$ is gradually updated from $\mathbf{E}^{\mathsf{f}}$ to $\mathbf{E}^{\mathsf{a}}$ using N_{y} local sequential updates.

4.3.1.4 The partition and the particle filter

So far, the sequential localisation framework looks elegant. The resulting generic analysis methods gradually update the ensemble $\mathbf{E}$ from $\mathbf{E}^{\mathsf{f}}$ to $\mathbf{E}^{\mathsf{a}}$. Furthermore, by using conditional local sequential updates, they have the potential to avoid the discontinuity issue inherent to the block localisation framework, and hence to mitigate the imbalance.

However, in a PF context, where the pdfs are approximated by sums of Dirac kernels, non-zero factors in $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ can only be avoided if the posterior particles are copies of the prior particles. This would spoil the entire purpose of localisation, and this is why potential solutions need to make approximations of $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$.

4.3.1.5 The multivariate rank histogram filter

Similar principles have been used to design the multivariate rank histogram filter (MRHF) algorithm of Metref et al. (2014), with the main difference being that the state space $\mathbb{R}^{N_{\mathrm{x}}}$ is entirely partitioned as follows. Assuming that the q-th observation $\boldsymbol{y}_q$ only depends on the first element of the state vector $\boldsymbol{x}_1$, $\pi[\boldsymbol{x}|\boldsymbol{y}_q]$ can be written

$$\pi[\boldsymbol{x}|\boldsymbol{y}_q] = \pi[\boldsymbol{x}_1|\boldsymbol{y}_q] \prod_{n=1}^{N_{\mathrm{x}}-1} \pi[\boldsymbol{x}_{n+1}|\boldsymbol{x}_{n:1}]. \tag{4.52}$$

Algorithm 4.6: Analysis step for a generic LPF–Y algorithm using the UVW partition.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $\mathbf{y}$
Parameters: $\pi^{\circ}$, $\boldsymbol{\rho}$
$\mathbf{E} \leftarrow \mathbf{E}^{\mathsf{f}}$                              // initialise the sequential updates
for $q = 1$ to $N_{\mathrm{y}}$ do
    From $\boldsymbol{\rho}$, build the UVW partition  // as described in paragraph 4.3.1.1
    for $i = 1$ to $N_{\mathrm{e}}$ do
        Do not update $\mathbf{x}_{\mathsf{W}}(i)$
        Update $\mathbf{x}_{\mathsf{U}}(i)$ conditionally to $y_q$
        Update $\mathbf{x}_{\mathsf{V}}(i)$ conditionally to $\mathbf{x}_{\mathsf{W}}(i)$ and (the updated) $\mathbf{x}_{\mathsf{U}}(i)$
    end
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \mathbf{E}$
Output: $\mathbf{E}^{\mathsf{a}}$
```

In the analysis step of the MRHF algorithm, the state variables are updated sequentially according to the densities $\pi[\boldsymbol{x}_{n+1}|\boldsymbol{x}_{n:1}], n \in (N_{\mathrm{x}} - 1 : 1)$. Zero factors in the n-th conditional density $\pi[\boldsymbol{x}_{n+1}|\boldsymbol{x}_{n:1}]$ are avoided by using a kernel representation for the conditioning with respect to $\boldsymbol{x}_{n:1}$, in a similar way as in equations (4.35) and (4.36), but using top-hat functions for the regularisation kernel $\mathcal{K}$. The resulting one-dimensional density along the $(n+1)$-th variable $\boldsymbol{x}_{n+1}$ is represented using histograms, and the particles are transformed using the anamorphosis method, as described in subsection 4.2.3.

The MRHF could be used as a potential implementation of the block localisation framework. However, assimilating only one observation requires the computation of N_{x} different anamorphosis transformations.

4.3.1.6 Towards an implementation of the sequential localisation framework

In the following subsections, we introduce two different algorithms which implement the sequential localisation framework with the UVW partition. Both algorithms are based on an *importance, resampling, propagation* scheme as follows. In order to assimilate the q-th observation y_q, we first compute a global importance weight vector $\mathbf{w}$ using

$$\mathbf{w} = \pi^{\mathsf{o}}_q(y_q|\mathbf{E}), \tag{4.53}$$

where $\mathbf{E}$ denotes here the current ensemble, that is the ensemble after the assimilation of the first $q-1$ observations. Using $\mathbf{w}$, a resampling map ψ is computed and applied to the U region (essentially at the q-th observation site). This update is then propagated to the V region using a dedicated propagation algorithm.

4.3.2 An hybrid method for the propagation

In this subsection, we describe how the LPF algorithm of Poterjoy (2016) implements the sequential localisation framework. In order to simplify the presentation, we only describe the assimilation of the q-th observation y_q. Furthermore, the ensemble $\mathbf{E}$ before the assimilation of y_q (and hence after the assimilation of the first $q-1$ observations) is called here the *prior* ensemble, and written $\mathbf{E}^{\mathsf{f}}$, and the ensemble $\mathbf{E}$ after the assimilation of y_q is called here the *posterior* ensemble, and written $\mathbf{E}^{\mathsf{a}}$.

4.3.2.1 First step: importance and resampling

First, a global importance weight vector $\mathbf{w}$ is computed using

$$\mathbf{w} = \pi^{\mathsf{o}}_q\big(y_q|\mathbf{E}^{\mathsf{f}}\big). \tag{4.54}$$

In a second time, $\mathbf{w}$ is used to compute a resampling map ψ using, for example, the systematic resampling algorithm.

4.3.2.2 Second step: propagation

The resampling described by ψ is then propagated using an hybrid method mixing the global PF update and the prior ensemble $\mathbf{E}^{\mathsf{f}}$. With this method, the n-th variable of the i-th posterior particle $\mathbf{x}^{\mathsf{a}}(i)$ is obtained as

$$x^{\mathsf{a}}_n(i) = \bar{x}^{\mathsf{a}}_n + \omega^{\mathsf{a}}_n \Big[x^{\mathsf{f}}_n\big(\psi(i)\big) - \bar{x}^{\mathsf{a}}_n \Big] + \omega^{\mathsf{f}}_n \Big[x^{\mathsf{f}}_n(i) - \bar{x}^{\mathsf{a}}_n \Big], \tag{4.55}$$

where $\bar{x}^{\mathsf{a}}_n$ is the posterior mean of the n-th variable, defined using the local importance weight vector $\mathbf{w}_n$ as

$$\bar{x}^{\mathsf{a}}_n \triangleq \sum_{i=1}^{N_\mathrm{e}} \bar{w}_n(i)\, x^{\mathsf{f}}_n(i). \tag{4.56}$$

In equation (4.55), the prior and posterior **update weights** ω^{f}_n and ω^{a}_n control the magnitude of the PF update. They are chosen in such a way that $\mathbf{E}^{\mathsf{a}}$ yields correct statistics at the first order:

$$\frac{1}{N_\mathrm{e}} \sum_{i=1}^{N_\mathrm{e}} x^{\mathsf{a}}_n(i) = \bar{x}^{\mathsf{a}}_n, \tag{4.57}$$

and at the second order:

$$\frac{1}{N_\mathrm{e}-1} \sum_{i=1}^{N_\mathrm{e}} \big(x^{\mathsf{a}}_n(i) - \bar{x}^{\mathsf{a}}_n \big)^2 = (\sigma^{\mathsf{a}}_n)^2, \tag{4.58}$$

where $(\sigma^{\mathsf{a}}_n)^2$ is the posterior variance of the n-th variable, defined by

$$(\sigma^{\mathsf{a}}_n)^2 \triangleq \frac{1}{1 - \bar{\mathbf{w}}_n^{\mathsf{T}} \bar{\mathbf{w}}_n} \sum_{i=1}^{N_\mathrm{e}} \bar{w}_n(i) \big(x^{\mathsf{f}}_n(i) - \bar{x}^{\mathsf{a}}_n \big)^2. \tag{4.59}$$

As shown by Poterjoy (2016), if the $\mathbf{w}_{N_\mathrm{x}:1}$ are computed using the generic formulation, equation (4.22), then a solution to this problem is to use

$$c_n = \frac{\alpha_q N_\mathrm{e} \left[1 - G\left(\dfrac{2d_{q,n}}{\ell} \right) \right]}{G\left(\dfrac{2d_{q,n}}{\ell} \right) \displaystyle\sum_{i=1}^{N_\mathrm{e}} w(i)}, \tag{4.60}$$

$$\omega^{\mathsf{a}}_n = \frac{\sigma^{\mathsf{a}}_n}{\left[\dfrac{1}{N_\mathrm{e}-1} \displaystyle\sum_{i=1}^{N_\mathrm{e}} \Big[x^{\mathsf{f}}_n\big(\psi(i)\big) - \bar{x}^{\mathsf{a}}_n + c_n \big\{ x^{\mathsf{f}}_n(i) - \bar{x}^{\mathsf{a}}_n \big\} \Big]^2 \right]^{1/2}}, \tag{4.61}$$

$$\omega^{\mathsf{a}}_n = c_n \omega^{\mathsf{a}}_n, \tag{4.62}$$

where $w(i)$ is the i-th element of the global importance weight vector $\mathbf{w}$ given by equation (4.54), and where $d_{q,n}$ is the distance between the q-th site and the n-th grid point.

At the q-th site (*i.e.*, when $d_{q,n} \to 0$) $\omega^{\mathsf{f}}_n = 0$ and $\omega^{\mathsf{a}}_n = 1$, and equation (4.55) matches the global PF update. Far from the q-th site (*i.e.*, when $d_{q,n} \geq \ell$) $\omega^{\mathsf{f}}_n = 1$ and $\omega^{\mathsf{a}}_n = 0$, and there is no update. Therefore, the i-th updated particle $\mathbf{x}^{\mathsf{a}}(i)$ is a composite particle between the i-th prior particle $\mathbf{x}^{\mathsf{f}}(i)$, in the W region, and the hypothetical i-th updated particle $\mathbf{x}^{\mathsf{f}}\big(\psi_q(i)\big)$ which would be obtained with the global PF, in the U region. In-between, in the V region,

Algorithm 4.7: Single analysis step for a generic LPF–Y algorithm using the hybrid propagation method.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $y_q$
Parameters: $\pi^{\mathsf{o}}_q$, $\ell$, $\alpha_q$
w ← $\pi^{\mathsf{o}}_q\big(y_q|\mathbf{E}^{\mathsf{f}}\big)$
$\psi$ ← ResamplingMap$\big(\mathbf{w},\mathbf{E}^{\mathsf{f}}\big)$
for $n = 1$ to $N_{\mathrm{x}}$ do
    $\omega^{\mathsf{a}}$ ← equation (4.61)
    $\omega^{\mathsf{f}}$ ← equation (4.62)
    for $i = 1$ to $N_{\mathrm{e}}$ do
        $x^{\mathsf{a}}_n(i) \leftarrow \bar{x}^{\mathsf{a}}_n + \omega^{\mathsf{a}}_n\Big[x^{\mathsf{f}}_n\big(\psi(i)\big) - \bar{x}^{\mathsf{a}}_n\Big] + \omega^{\mathsf{f}}_n\Big[x^{\mathsf{f}}_n(i) - \bar{x}^{\mathsf{a}}_n\Big]$
    end
end
$\mathbf{E}^{\mathsf{a}} \leftarrow \mathbf{E}$
Output: $\mathbf{E}^{\mathsf{a}}$
```

discontinuities are avoided by using a smooth transition for the prior and posterior update weights ω^{f} and ω^{a}, as described by equations (4.61) and (4.62). Algorithm 4.7 summarises the assimilation of y_q for a generic LPF–Y algorithm using the propagation method of Poterjoy (2016), hereafter called the **hybrid propagation method**.

In his original algorithm, Poterjoy (2016) included a weight inflation parameter which can be ignored to understand how the algorithm works. Moreover, the N_{y} local sequential updates are followed by an optional KDDM step. As explained in paragraph 4.2.3.5, we found the KDDM step to be better suited for the local update of the LPF–X algorithms. Therefore, we have not included these elements in our presentation the LPF algorithm of Poterjoy (2016).

4.3.3 A second-order method for the propagation

In this subsection, we introduce a new method, in which the update is propagated using second-order statistical moments. This method is inspired from the EnKPF algorithm, a Gaussian mixture hybrid ensemble filter designed by S. Robert and Künsch (2017). For simplicity, the notation introduced in the previous subsection is kept.

4.3.3.1 The prior covariance matrix

Because the update is propagated using second-order statistical moments, we first need to compute the sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$ of the prior ensemble $\mathbf{E}^{\mathsf{f}}$. Moreover, in a localisation context, it seems reasonable to use a localised representation of the covariance, as introduced in subsection 2.5.3 for CL in the EnKF. Therefore, $\bar{\mathbf{P}}^{\mathsf{f}}$ is here computed using

$$\mathbf{X} = \mathbf{E}^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}, \tag{4.63}$$

$$\bar{\mathbf{P}}^{\mathsf{f}} = \boldsymbol{\rho} \circ \big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big), \tag{4.64}$$

where $\boldsymbol{\rho}$ is the localisation matrix introduced in paragraph 4.3.1.1 for the definition of the partition.

4.3.3.2 First step: importance and resampling

As in the first step of the LPF–Y algorithm with the hybrid propagation method, described in paragraph 4.3.2.1, we first compute a global importance weight vector $\mathbf{w}$ and a resampling map ψ. For each particle $i \in (N_{\mathrm{e}} : 1)$, we then compute the update on the U region as

$$\Delta\mathbf{x}_{\mathsf{U}}(i) \triangleq \mathbf{x}^{\mathsf{f}}_{\mathsf{U}}\big(\psi(i)\big) - \mathbf{x}^{\mathsf{f}}_{\mathsf{U}}(i). \tag{4.65}$$

4.3.3.3 Second step: propagation

For each particle $i \in (N_{\mathrm{e}} : 1)$, the update on the U region, $\Delta\mathbf{x}_{\mathsf{U}}(i)$, is propagated to the V region through the linear regression

$$\Delta\mathbf{x}_{\mathsf{V}}(i) = \bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{VU}}\big(\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{U}}\big)^{-1}\Delta\mathbf{x}_{\mathsf{U}}(i), \tag{4.66}$$

where $\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{VU}}$ and $\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{U}}$ are the sub-matrices of $\bar{\mathbf{P}}^{\mathsf{f}}$ corresponding to the U and V regions. The full derivation of equation (4.66) can be found in S. Robert and Künsch (2017).

Finally, the update on the U and V regions is applied to obtain the posterior ensemble. Algorithm 4.8 summarises the assimilation of the q-th observation y_q for a generic LPF–Y algorithm using this **second-order propagation method**.

4.3.3.4 Generalisation of the first step

Algorithm 4.8 can be straightforwardly generalised to the case where the resampling step on the U region, described in paragraph 4.3.3.2, is replaced by a linear transformation or transport step, following the method described in paragraphs 4.2.3.4 and 4.2.3.5 in the context of the block localisation framework.

4.4 Highlights

We conclude this chapter by providing, in subsection 4.4.1, a short summary of the LPF algorithms. Furthermore, in subsections 4.4.2 and 4.4.3, we study the algorithmic complexity of the LPF algorithms, and in subsections 4.4.4 and 4.4.5, we study the behaviour of the LPF algorithms in the limit of an infinite localisation radius, $\ell \to \infty$.

Algorithm 4.8: Single analysis step for a generic LPF–Y algorithm using the second-order propagation method.

```
Input: $\mathbf{E}^{\mathsf{f}}$, $y_q$
Parameters: $\pi^{\mathsf{o}}_q$, $\boldsymbol{\rho}$
From $\boldsymbol{\rho}$, build the UVW partition        // as described in paragraph 4.3.1.1
$\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}$
$\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{U}} \leftarrow \boldsymbol{\rho}_{\mathsf{U}} \circ \big(\mathbf{X}_{\mathsf{U}}\mathbf{X}^{\mathsf{T}}_{\mathsf{U}}\big)$
$\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{VU}} \leftarrow \boldsymbol{\rho}_{\mathsf{VU}} \circ \big(\mathbf{X}_{\mathsf{V}}\mathbf{X}^{\mathsf{T}}_{\mathsf{U}}\big)$
$\mathbf{w} \leftarrow \pi^{\mathsf{o}}_q\big(y_q|\mathbf{E}^{\mathsf{f}}\big)$
$\psi \leftarrow \texttt{ResamplingMap}\big(\mathbf{w}, \mathbf{E}^{\mathsf{f}}\big)$
for $i = 1$ to $N_{\mathrm{e}}$ do
    $\Delta\mathbf{x}_{\mathsf{U}}(i) \leftarrow \mathbf{x}^{\mathsf{f}}_{\mathsf{U}}\big(\psi(i)\big) - \mathbf{x}^{\mathsf{f}}_{\mathsf{U}}(i)$
    $\Delta\mathbf{x}_{\mathsf{V}}(i) \leftarrow \bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{VU}}\big(\bar{\mathbf{P}}^{\mathsf{f}}_{\mathsf{U}}\big)^{-1}\Delta\mathbf{x}_{\mathsf{U}}(i)$
    $\mathbf{x}^{\mathsf{a}}_{\mathsf{U}}(i) \leftarrow \mathbf{x}^{\mathsf{f}}_{\mathsf{U}}(i) + \Delta\mathbf{x}_{\mathsf{U}}(i)$
    $\mathbf{x}^{\mathsf{a}}_{\mathsf{V}}(i) \leftarrow \mathbf{x}^{\mathsf{f}}_{\mathsf{V}}(i) + \Delta\mathbf{x}_{\mathsf{V}}(i)$
    $\mathbf{x}^{\mathsf{a}}_{\mathsf{W}}(i) \leftarrow \mathbf{x}^{\mathsf{f}}_{\mathsf{W}}(i)$
end
Output: $\mathbf{E}^{\mathsf{a}}$
```

4.4.1 Summary: the LPF–X and LPF–Y algorithms

In section 4.2, a generic block localisation framework has been constructed to define the LPF–X algorithms. The LPF–X algorithms are characterised by the geometry of the local blocks and domains – in other words, by the definition of the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$ – and by the method used to locally update the prior ensemble $\mathbf{E}^\mathsf{f}$. As shown by Rebeschini and van Handel (2015), the LPF–X algorithms can beat the curse of dimensionality. However, unphysical discontinuities are likely to arise after assembling the locally updated ensembles, which can yield imbalance (van Leeuwen 2009). Four approaches have been proposed to mitigate such discontinuities.

1. A smoothing-by-weights step can be applied after the local update in order to reduce potential unphysical discontinuities. The method presented in the subsection 4.2.3.1 is a generalisation of the original smoothing designed by Penny and Miyoshi (2016) which is suited to the use of local state blocks, and which includes a spatial tapering controlled by the smoothing radius ℓ^s, and a smoothing strength parameter α^s.

2. Simple properties of the local resampling algorithms can be used in order to minimise the occurrences of unphysical discontinuity as shown by S. Robert and Künsch (2017).

3. Using the principles of optimal ensemble coupling, the local resampling can be replaced by a linear transformation step. The resulting algorithm is a local variant of the ETPF algorithm of Reich (2013), which can be seen as a generalisation of the algorithm of Cheng and Reich (2015) suited to the use of local state blocks. By construction, on each local block b, the distance between the local prior and posterior ensembles $\mathbf{E}^\mathsf{f}_{|b}$ and $\mathbf{E}^\mathsf{r}_{|b}$ is minimised.

4. By combining the continuous optimal transport problem with the KDE theory, a new local update method, based on the anamorphosis, has been derived. Furthermore, the properties of the anamorphosis have been shown to help mitigate the unphysical discontinuities.

In section 4.3, a generic sequential localisation framework has been constructed to define the LPF–Y algorithms. Two LPF–Y algorithms, both based on an *importance, resampling, propagation* scheme have been presented.

1. The first algorithm uses the hybrid propagation method derived by Poterjoy (2016), which mixes the prior ensemble $\mathbf{E}^\mathsf{f}$ with the global PF update to define the posterior ensemble $\mathbf{E}^\mathsf{a}$.

2. The second algorithm is inspired from the EnKPF algorithm of S. Robert and Künsch (2017). It uses a propagation method based on localised second-order statistical moments.

Both algorithms include some spatial smoothness in the construction of the posterior ensemble $\mathbf{E}^\mathsf{a}$. In the first method, the smoothness comes from the definition of $\omega^\mathsf{f}_{N_\mathrm{x}:1}$ and $\omega^\mathsf{a}_{N_\mathrm{x}:1}$. In the second method, the smoothness comes from the localised prior sample covariance matrix $\bar{\mathbf{P}}^\mathsf{f}$. Therefore, we expect the unphysical discontinuities, and hence the imbalance, to be less critical with these algorithms than with the LPF–X algorithms, which is why the partition was introduced in the first place.

4.4.2 Algorithmic complexity of the LPF–X algorithms

4.4.2.1 Auxiliary quantities

In order to study the algorithmic complexity of the LPF–X algorithms, we introduce the auxiliary quantities

$$N_{\mathrm{y}}^{\ell}(\ell) \triangleq \max_{b\in(N_{\mathrm{b}}:1)} \mathrm{Card}\Big\{ q \in (N_{\mathrm{y}}:1) \setminus d_{q,b} \leq \ell \Big\}, \tag{4.67}$$

$$N_{\mathrm{x}}^{\ell}(\ell^{\mathsf{d}}) \triangleq \max_{b\in(N_{\mathrm{b}}:1)} \mathrm{Card}\Big\{ n \in (N_{\mathrm{x}}:1) \setminus d_{n,b} \leq \ell^{\mathsf{d}} \Big\}, \tag{4.68}$$

$$N_{\mathrm{b}}^{\ell}(\ell^{\mathsf{s}}) \triangleq \max_{n\in(N_{\mathrm{x}}:1)} \mathrm{Card}\Big\{ b \in (N_{\mathrm{b}}:1) \setminus d_{n,b} \leq \ell^{\mathsf{s}} \Big\}. \tag{4.69}$$

The quantity $N_{\mathrm{y}}^{\ell}(\ell)$ is the maximum number of observations in a given local domain. It is a generalisation of the definition provided in paragraph 2.5.4.1 which includes the concept of local blocks. The quantity $N_{\mathrm{x}}^{\ell}(\ell^{\mathsf{d}})$ is the maximum number of variables whose grid point is located within distance ℓ^{d} to the centre of a given local block. The quantity $N_{\mathrm{b}}^{\ell}(\ell^{\mathsf{s}})$ is the maximum number of local blocks whose centre is located within distance ℓ^{s} to a given grid point.

4.4.2.2 Algorithmic complexity of computing the local importance weight vectors

When the observations are independent, the local importance weight vectors $\mathbf{w}_{N_{\mathrm{b}}:1}$ are given by equation (4.22) for the generic formulation, or by equation (4.25) for the Gaussian formulation. In both cases, the algorithmic complexity is

$$\mathcal{O}\big(N_{\mathrm{e}}T_{\mathcal{H}} + N_{\mathrm{b}}N_{\mathrm{e}}N_{\mathrm{y}}^{\ell}(\ell)\big),$$

which depends on the localisation radius ℓ, and on the algorithmic complexity $T_{\mathcal{H}}$ of applying the observation operator $\mathcal{H}$ to a vector.

When the observations are not independent, the $\mathbf{w}_{N_{\mathrm{b}}:1}$ are given by equation (4.24) for the Gaussian formulation. In this case, the algorithmic complexity is

$$\mathcal{O}\big(N_{\mathrm{e}}T_{\mathcal{H}} + N_{\mathrm{b}}N_{\mathrm{e}}[N_{\mathrm{y}}^{\ell}(\ell)]^2\big).$$

4.4.2.3 Algorithmic complexity of the local updates using resampling

When the local updates are performed with the multinomial or the systematic resampling algorithm, the algorithmic complexity of each local update is

- $\mathcal{O}(N_{\mathrm{e}})$ to compute the resampling map ψ;
- $\mathcal{O}(N_{\mathrm{e}})$ per state variable to apply the resampling map ψ.

Therefore, the total complexity of the local updates is $\mathcal{O}(N_{\mathrm{x}}N_{\mathrm{e}})$.

4.4.2.4 Algorithmic complexity of the local updates using optimal ensemble coupling

When using optimal ensemble coupling for the local updates, the total algorithmic complexity is higher, because we need to solve one optimisation problem per local block. The algorithmic complexity of each local update is the sum of three terms.

- The algorithmic complexity of computing the local cost coefficients $\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big)$, given by equation (4.32), is $\mathcal{O}\big(N_{\mathrm{e}}^2 N_{\mathrm{x}}^{\ell}(\ell^{\mathsf{d}})\big)$, which depends on the distance radius ℓ^{d}.

- The optimisation problem for the b-th local block consists in minimising the cost function defined by equation (4.31) under the constraint $\mathbf{T}_{\mathrm{e}} \in \mathbb{T}(\mathbf{w}_b)$. This is a particular case of the minimum-cost flow problem, and hence it can be solved quite efficiently using the algorithm of Pele and Werman (2009) with algorithmic complexity $\mathcal{O}\big(N_{\mathrm{e}}^2 \ln N_{\mathrm{e}}\big)$.

- Applying the linear transformation to the local block has algorithmic complexity $\mathcal{O}\big(N_{\mathrm{e}}^2\big)$ per state variable.

Therefore, the total algorithmic complexity of the local updates is

$$\mathcal{O}\big(N_{\mathrm{b}} N_{\mathrm{e}}^2 N_{\mathrm{x}}^{\ell}(\ell^{\mathsf{d}}) + N_{\mathrm{b}} N_{\mathrm{e}}^2 \ln N_{\mathrm{e}} + N_{\mathrm{x}} N_{\mathrm{e}}^2\big).$$

4.4.2.5 Algorithmic complexity of the local updates using anamorphosis

When using anamorphosis for the local updates, as presented in paragraph 4.2.3.5, every one-dimensional transformation is computed with algorithmic complexity $\mathcal{O}(N_{\mathrm{p}})$, where N_{p} is the one-dimensional resolution for each state variable. Therefore, the total algorithmic complexity of the local updates is

$$\mathcal{O}(N_{\mathrm{x}} N_{\mathrm{e}} N_{\mathrm{p}}).$$

4.4.2.6 Algorithmic complexity of the additional smoothing-by-weights step

When using the smoothing-by-weights step with the multinomial or the systematic resampling algorithm (as presented in paragraph 4.2.3.2), the smoothed ensemble $\mathbf{E}^{\mathsf{s}}$ is computed with algorithmic complexity $\mathcal{O}\big(N_{\mathrm{x}} N_{\mathrm{e}} N_{\mathrm{b}}^{\ell}(\ell^{\mathsf{s}})\big)$, which depends on the smoothing radius ℓ^{s}, and the posterior ensemble $\mathbf{E}^{\mathsf{a}}$ is computed with algorithmic complexity $\mathcal{O}(N_{\mathrm{x}} N_{\mathrm{e}})$. Therefore, the total algorithmic complexity of the resampling and the smoothing steps is

$$\mathcal{O}\big(N_{\mathrm{x}} N_{\mathrm{e}} N_{\mathrm{b}}^{\ell}(\ell^{\mathsf{s}})\big).$$

4.4.2.7 Summary table

The algorithmic complexity for each step of the LPF–X algorithms is summarised in table 4.1. Finally, the local updates are by construction embarrassingly parallel. Therefore, the total complexity of the analysis step of the LPF–X algorithms can be reduced by a factor N_{t}, the number of threads running in parallel.

Table 4.1: Summary table for the algorithmic complexity of the LPF–X algorithms.

Compute the local importance weight vectors	
Diagonal case	$\mathcal{O}\big(N_{\mathrm{e}}T_{\mathcal{H}} + N_{\mathrm{b}}N_{\mathrm{e}}N^{\ell}_{\mathrm{y}}(\ell)\big)$
Non-diagonal case – Gaussian form	$\mathcal{O}\big(N_{\mathrm{e}}T_{\mathcal{H}} + N_{\mathrm{b}}N_{\mathrm{e}}[N^{\ell}_{\mathrm{y}}(\ell)]^2\big)$
Perform the local updates – multinomial or systematic resampling	
Compute the resampling maps	$\mathcal{O}(N_{\mathrm{b}}N_{\mathrm{e}})$
Apply the resampling maps	$\mathcal{O}(N_{\mathrm{x}}N_{\mathrm{e}})$
Perform the local updates – optimal ensemble coupling	
Compute the local cost coefficients	$\mathcal{O}\big(N_{\mathrm{b}}N_{\mathrm{e}}^2N^{\ell}_{\mathrm{x}}(\ell^{\mathsf{d}})\big)$
Compute the optimal LET matrices	$\mathcal{O}\big(N_{\mathrm{b}}N_{\mathrm{e}}^2 \ln N_{\mathrm{e}}\big)$
Apply the linear transformations	$\mathcal{O}\big(N_{\mathrm{x}}N_{\mathrm{e}}^2\big)$
Perform the local updates – anamorphosis	
Compute and apply the transformation	$\mathcal{O}(N_{\mathrm{x}}N_{\mathrm{e}}N_{\mathrm{p}})$
Additional smoothing-by-weights step	
Compute the smoothed ensemble	$\mathcal{O}\big(N_{\mathrm{x}}N_{\mathrm{e}}N^{\ell}_{\mathrm{b}}(\ell^{\mathsf{s}})\big)$
Compute the posterior ensemble	$\mathcal{O}(N_{\mathrm{x}}N_{\mathrm{e}})$

4.4.3 Algorithmic complexity of the LPF–Y algorithms

4.4.3.1 Auxiliary quantities

In order to study the algorithmic complexity of the LPF–Y algorithms, we introduce the auxiliary quantities N_{U} and N_{V}, defined as the maximum number of state variables in the U and V regions, and $N_{\mathsf{UV}} = N_{\mathsf{U}} + N_{\mathsf{V}}$. Furthermore, the quantity $N^{\ell}_{\mathrm{x}}(\ell^{\mathsf{d}})$ is now defined as

$$N^{\ell}_{\mathrm{x}}(\ell^{\mathsf{d}}) \triangleq \max_{q\in(N_{\mathrm{y}}:1)} \mathrm{Card}\Big\{n \in (N_{\mathrm{x}}:1) \setminus d_{n,q} \leq \ell^{\mathsf{d}}\Big\}. \tag{4.70}$$

In other words, $N^{\ell}_{\mathrm{x}}(\ell^{\mathsf{d}})$ is the maximum number of variables whose grid point is located within distance ℓ^{d} to the site of a given observation.

4.4.3.2 Algorithmic complexity for the hybrid propagation method

When using the hybrid propagation method, the algorithmic complexity of assimilating the q-th observation y_q is

- $\mathcal{O}(N_{\mathrm{e}})$ to compute the global importance weight vector $\mathbf{w}$ and the resampling map ψ;
- $\mathcal{O}(N_{\mathrm{e}})$ per state variable to compute the prior and posterior update weights ω^{f} and ω^{a}, and to compute the posterior ensemble $\mathbf{E}^{\mathsf{a}}$.

Therefore, the total algorithmic complexity of each local sequential update is $\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{UV}})$.

4.4.3.3 Algorithmic complexity for the second-order propagation method

When using the second-order propagation method, the algorithmic complexity of assimilating the q-th observation y_q is

- $\mathcal{O}(N_\mathrm{e}N_\mathsf{U})$ to compute the update in the U region with the multinomial or the systematic resampling algorithm;
- $\mathcal{O}\big(N_\mathrm{e}^2 N_\mathrm{x}^\ell(\ell^\mathsf{d}) + N_\mathrm{e}^2 \ln N_\mathrm{e} + N_\mathsf{U} N_\mathrm{e}^2\big)$ to compute the update in the U region with the optimal ensemble coupling;
- $\mathcal{O}(N_\mathrm{e}N_\mathsf{U}N_\mathrm{p})$ to compute the update in the U region with the anamorphosis using a fixed one-dimensional resolution of N_p points;
- $\mathcal{O}\big(N_\mathsf{U}^3\big)$ to compute the inverse of the $N_\mathsf{U} \times N_\mathsf{U}$-matrix $\bar{\mathbf{P}}_\mathsf{U}^\mathsf{f}$;
- $\mathcal{O}\big(N_\mathrm{e}N_\mathsf{U}^2 + N_\mathrm{e}N_\mathsf{V}N_\mathsf{U}\big)$ to compute the update on the V region using the linear regression given by equation (4.66).
- $\mathcal{O}(N_\mathrm{e}N_\mathsf{UV})$ to apply the update to the N_UV variables in the U and V regions.

4.4.3.4 Summary table

The algorithmic complexity for each step of the LPF–Y algorithms is summarised in table 4.2. By construction, the N_y local sequential updates are not parallel. This issue is discussed by S. Robert and Künsch (2017): some level of parallelisation could be introduced in the algorithms, but only between observations for which the U and V regions are disjoint. That is to say, one can assimilate the observations at several sites in parallel as long as their domains of influence (in which an update is needed) do not overlap. This would require a preliminary geometric step to determine the order in which the observations are to be assimilated. This step would need to be performed again whenever the localisation radius ℓ is changed. Moreover, when ℓ is large enough, all U and V regions may overlap, and parallelisation is not possible.

4.4.4 Asymptotic limit of the LPF–X algorithms

As discussed in section 3.5, under minimal conditions, the empirical analysis density $\bar{\pi}^\mathsf{a}$ of the PF weakly converges towards the analysis density π^a in the limit of an infinite ensemble, $N_\mathrm{e} \to \infty$. In the limit of an infinite localisation radius, $\ell \to \infty$, the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$, as defined by equation (4.22) or equation (4.24), all converge towards the global importance weight vector $\mathbf{w}$, as defined by equation (4.21). However, this does not necessarily imply that the resulting LPF–X assimilation cycle is equivalent to a global PF assimilation cycle, precisely because the local updates are by construction independent. In the following paragraphs, we study under which conditions the LPF–X assimilation cycle can nevertheless be equivalent to a global PF assimilation cycle.

Table 4.2: Summary table for the algorithmic complexity of the LPF–Y algorithms. For a full assimilation cycle (N_{y} local sequential updates), the algorithmic complexity is increased by a factor N_{y}.

With the hybrid propagation method	
Compute the resampling map	$\mathcal{O}(N_{\mathrm{e}})$
Propagate the update to the U and V regions	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{UV}})$
With the second-order propagation method – resampling	
Compute the update in the U region	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{U}})$
Compute the update in the V region	$\mathcal{O}\big(N_{\mathsf{U}}^3 + N_{\mathrm{e}}N_{\mathsf{U}}^2 + N_{\mathrm{e}}N_{\mathsf{V}}N_{\mathsf{U}}\big)$
Apply the update in the U and V regions	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{UV}})$
With the second-order propagation method – optimal ensemble coupling	
Compute the update in the U region	$\mathcal{O}\big(N_{\mathrm{e}}^2 N_{\mathrm{x}}^{\ell}(\ell^{\mathsf{d}}) + N_{\mathrm{e}}^2 \ln N_{\mathrm{e}} + N_{\mathsf{U}}N_{\mathrm{e}}^2\big)$
Compute the update in the V region	$\mathcal{O}\big(N_{\mathsf{U}}^3 + N_{\mathrm{e}}N_{\mathsf{U}}^2 + N_{\mathrm{e}}N_{\mathsf{V}}N_{\mathsf{U}}\big)$
Apply the update in the U and V regions	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{UV}})$
With the second-order propagation method – anamorphosis	
Compute the update in the U region	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{U}}N_{\mathrm{p}})$
Compute the update in the V region	$\mathcal{O}\big(N_{\mathsf{U}}^3 + N_{\mathrm{e}}N_{\mathsf{U}}^2 + N_{\mathrm{e}}N_{\mathsf{V}}N_{\mathsf{U}}\big)$
Apply the update in the U and V regions	$\mathcal{O}(N_{\mathrm{e}}N_{\mathsf{UV}})$

4.4.4.1 Asymptotic limit for the algorithms using resampling

When the local updates are performed with the multinomial or the systematic resampling algorithm, the LPF–X assimilation cycle is equivalent to the assimilation cycle of the SIR algorithm, algorithm 3.2, in the limit of an infinite localisation radius, $\ell \to \infty$, if, and only if the same random number(s) are used in the local resampling of all local blocks.

In all other cases, the posterior ensemble $\mathbf{E}^{\mathsf{a}}$ is by construction distributed according to the product of the marginals of the empirical analysis density $\bar{\pi}^{\mathsf{a}}_{N_{\mathrm{x}}:1}$, given by equation (4.20), which is in general different from the empirical analysis density $\bar{\pi}^{\mathsf{a}}$ of the PF, even in the limit of an infinite localisation radius, $\ell \to \infty$, and of an infinite ensemble, $N_{\mathrm{e}} \to \infty$.

4.4.4.2 Asymptotic limit for the algorithms using optimal ensemble coupling

The local cost coefficients $\mathcal{C}_b\big(\mathbf{E}^{\mathsf{f}}\big)$, given by equation (4.32), converges towards the global cost coefficients of the global ETPF algorithm, algorithm 3.5, deduced from equation (3.61), in the limit of an infinite distance radius, $\ell^{\mathsf{d}} \to \infty$. Moreover, in the limit of an infinite localisation radius, $\ell \to \infty$, the $\mathbf{w}_{N_{\mathrm{b}}:1}$ converge towards $\mathbf{w}$, in which case the constraints of the local optimisation problems are equivalent to those of the global optimisation problem in the ETPF algorithm. Therefore, when using optimal ensemble coupling for the local updates, the LPF–X assimilation cycle is equivalent to the assimilation cycle of the ETPF algorithm in the limit of an infinite localisation radius, $\ell \to \infty$, and an infinite distance radius, $\ell^{\mathsf{d}} \to \infty$.

4.4.4.3 Asymptotic limit for the algorithms using anamorphosis

When using anamorphosis for the local updates the transport condition, equation (4.33), is defined independently for each state variable. Therefore, there is no proof that the LPF–X update follows Bayes' theorem, even in the limit of a zero forecast and analysis regularisation bandwidths, $h^{\mathsf{f}} \to 0$ and $h^{\mathsf{a}} \to 0$, of an infinite localisation radius, $\ell \to \infty$, and of an infinite ensemble, $N_{\mathrm{e}} \to \infty$.

4.4.4.4 Asymptotic limit for the algorithms using the smoothing-by-weights step

When using the smoothing-by-weights step with the multinomial or the systematic resampling algorithm, the smoothed ensemble $\mathbf{E}^{\mathsf{s}}$ is equal to the locally resampled ensemble $\mathbf{E}^{\mathsf{r}}$ in the limit of an infinite localisation radius, $\ell \to \infty$, if, and only if the same random number(s) are used in the local resampling of all local blocks. In this case, the LPF–X assimilation cycle is equivalent to the assimilation cycle of the SIR algorithm. In all other cases, we cannot give a firm answer, for the exact same reasons as in paragraph 4.4.4.1.

4.4.5 Asymptotic limit of the LPF–Y algorithms

By construction of the prior and posterior update weights ω^{f} and ω^{a}, assimilating the q-th observation $\mathbf{y}_q$ with the LPF–Y algorithm using the hybrid propagation method is equivalent to assimilating $\mathbf{y}_q$ with the SIR algorithm in the limit of an infinite localisation radius, $\ell \to \infty$. Therefore, a full assimilation cycle of the LPF–Y algorithm based on the hybrid propagation method is equivalent to a sequential version of the SIR algorithm in the limit of an infinite localisation radius, $\ell \to \infty$.

By contrast, this is not the case for the LPF–Y algorithm based on the second-order propagation method. Indeed, in general, using second-order moments to propagate the update introduces a bias in the analysis.

4.5 Summary and discussion

The curse of dimensionality is a rather well-understood phenomenon in the statistical literature, and it is the main reason why the PF fails in high-dimensional DA systems. In this chapter, we have recalled the main results related to the weight degeneracy in the PF, and why the use of localisation can be used as a solution. Yet implementing localisation in the PF raises two major issues: how to glue together locally updated particles and how to avoid imbalance in the updated ensemble. Adequate solutions to these issues are not obvious, witness the few but dissimilar LPF algorithms developed in the geophysical literature. We have proposed a theoretical classification of LPF algorithms into two categories. For each category, we have presented the challenges of local particle filtering and have reviewed the ideas leading to practical implementation of LPF algorithms. Some of them, already in the literature, have been detailed and sometimes generalised, while others are new in this field and yield improvements in the design of LPF algorithms.

In the LPF–X algorithms, the analysis is localised by allowing the importance weight vector to vary over the grid points. We have shown that this yields an analysis density from which only the marginals are known. The (global) analysis ensemble is obtained by assembling the locally updated particles, and its quality directly depends on the regularity of the local update method. This is related to potential unphysical discontinuities, and hence imbalance, in the assembled particles. Therefore we have presented practical methods to improve the local updates by reducing the unphysical discontinuities.

In the LPF–Y algorithms, localisation is introduced more generally in the analysis density by the means of a partition. The goal of the partition is to build a framework for local particle filtering without the discontinuity issue inherent to the LPF–X algorithms. We have shown how two methods can be used as an implementation of this framework. Besides, we have emphasised the fact that with these methods, observations are, by construction, assimilated sequentially, which is a great disadvantage when the number of observations of the DA system is high.

5 Localisation in the particle filter: numerical illustrations

Contents

This chapter is a direct continuation of chapter 4. Following Farchi and Bocquet (2018), the LPF algorithms are implemented and tested using twin experiments. Section 5.1 presents the DA systems which are selected for these illustrations. The numerical experiments are then described in sections 5.2 and 5.3. Finally, conclusions are given in section 5.4.

5.1 Presentation of the numerical experiments

5.1.1 Twin experiments and performance criteria

In this chapter, the performance of the DA algorithms are illustrated using **twin experiments**. First, a synthetic trajectory $(\mathbf{x}_k^{\mathrm{t}})_{k\in N_{\mathrm{c}}:}$ is simulated for the truth. Then, a synthetic sequence $(\mathbf{y}_k)_{k\in N_{\mathrm{c}}:}$ is simulated for the observation vector. Finally, the observation vectors are used to perform N_{c} assimilation cycles with a given DA algorithm.

The main interest of this controlled context lies in the fact that the background, observation, and transition distributions ν^{b}, ν^{o}, and ν^{m} of the DA system are perfectly known, and these elements can be used as is in the DA algorithms. Since the truth $\mathbf{x}^{\mathsf{t}}$ is perfectly known, it can be used to measure the performance of the DA algorithm. The most widespread approach is to compute the root-mean-square error (RMSE) between the analysis estimate $\mathbf{x}^{\mathsf{a}}$ and the truth $\mathbf{x}^{\mathsf{t}}$, defined as

$$R_k \triangleq \left[\frac{1}{N_{\mathrm{x}}}\left\|\mathbf{x}_k^{\mathsf{a}} - \mathbf{x}_k^{\mathsf{t}}\right\|_2^2\right]^{1/2}. \tag{5.1}$$

This instantaneous RMSE is usually averaged over the time period $t \in (N_{\mathrm{c}} : N_{\mathrm{s}})$, where $t \in (N_{\mathrm{s}} :)$ is considered as a spin-up period, during which the influence of the initialisation is progressively eliminated. Under the assumption that the dynamical system is ergodic, the average over time is equivalent to an average over the probability space. The resulting performance score is independent of specific realisations of the model and observation errors $\boldsymbol{e}^{\mathsf{m}}$ and $\boldsymbol{e}^{\mathsf{o}}$ and is representative of the expected distance between $\mathbf{x}^{\mathsf{a}}$ and $\mathbf{x}^{\mathsf{t}}$. In this thesis, it is called the **time-average analysis RMSE**, or simply the RMSE score.

By construction, the RMSE score only directly measures the accuracy of the analysis estimate $\mathbf{x}^{\mathsf{a}}$. The primary goal in DA is precisely to estimate the truth $\mathbf{x}^{\mathsf{t}}$. However, as formalised in subsection 1.1.4, the ultimate goal in filtering DA should be to estimate the analysis density π^{a}. However, even with low-order DA systems, the exact π^{a} is usually unknown. Moreover, even if π^{a} would be exactly known, the definition and the computation of a distance in the space of the pdfs over the state space $\mathbb{R}^{N_{\mathrm{x}}}$ is challenging. By contrast, the RMSE score provides a clear and unique performance score, which can be used to rank different DA algorithms.

In ensemble DA methods, the ensemble $\mathbf{E}$ provides information beyond the estimate of the truth $\mathbf{x}^{\mathsf{t}}$. This information is *a priori* not reflected in the RMSE score. For this reason, complementary tools exist to measure the quality of an ensemble $\mathbf{E}$ such as, for example, the **rank histograms** or **Talagrand diagrams** (Anderson 1996; Hamill and Colucci 1997; Talagrand et al. 1997). The rank histogram can be defined, for a variable $n \in (N_{\mathrm{x}} : 1)$, as the empirical frequencies of the rank of $\mathbf{x}^{\mathsf{t}}$ in the ensemble $\mathbf{E}$. Ideally, $\mathbf{x}^{\mathsf{t}}$ and $\mathbf{E}$ should be drawn from the same distribution. This means that $\mathbf{x}^{\mathsf{t}}$ should be indistinguishable from the members of $\mathbf{E}$, and the histogram should be flat. Compared to the RMSE score, the rank histograms provide more information (one histogram per state variable) but they are harder to interpret, and they do not yield a clear ranking of different DA algorithms. Furthermore, for a sufficiently high number of assimilation cycles N_{c}, it seems likely that good RMSE scores can only be achieved with an ensemble $\mathbf{E}$ of good quality in the light of most other indicators. This is why in most numerical experiments in this thesis, we adopt the RMSE score as performance criterion.

The following subsections describe the DA systems selected for the experiments in this chapter.

5.1.2 The Lorenz 1996 model

5.1.2.1 The dynamical model

The Lorenz 1996 (L96) dynamical model (Lorenz and Emanuel 1998) is a low-order one-dimensional discrete chaotic model whose evolution is given by the following set of ordinary differential equations (ODE):

$$\forall n \in (N_\mathrm{x} : 1), \quad \frac{\mathrm{d}x_n}{\mathrm{d}t} = (x_{n+1} - x_{n-2})\, x_{n-1} - x_n + F, \tag{5.2}$$

where x_n is the n-th variable of the vector $\mathbf{x}$, and where the indices are to be understood with periodic boundary conditions: $x_{-1} = x_{N_\mathrm{x}-1}$, $x_0 = x_{N_\mathrm{x}}$, and $x_1 = x_{N_\mathrm{x}+1}$. The dimension of the state space N_x can take arbitrary values. These ODEs are integrated using a fourth-order Runge–Kutta method with an integration time step δt of 0.05 unit of time, and without model error.

5.1.2.2 The mildly nonlinear configuration

For the L96 model, we define a *mildly nonlinear* DA configuration as follows. The dimension of the state space is $N_\mathrm{x} = 40$ and the forcing term is $F = 8$. The resulting dynamics is chaotic, with a doubling time around 0.42 unit of time. The time interval between consecutive observations Δt is set to 0.05 unit of time. This is meant to represent 6 h of real time, and it corresponds to a model autocorrelation around 0.967. Finally, the observation vector $\mathbf{y}$ is computed from the truth $\mathbf{x}^\mathrm{t}$ using

$$\mathbf{y}_k = \mathbf{x}^\mathrm{t}_k + \mathbf{e}^\mathrm{o}_k, \quad \mathbf{e}^\mathrm{o}_k \sim \mathcal{N}[\mathbf{0}, \mathbf{I}], \tag{5.3}$$

where the individual observation variance (namely 1) is approximately one tenth of the typical variability of each state variable.

This configuration is widely used in the DA literature to asses the performances of the DA algorithms. The nonlinearities are weak and representative of the synoptic scale meteorology, and they only come from the integration of the ODEs defined by equation (5.2). As a consequence, the error distributions are close to Gaussian. Furthermore, the number of unstable and neutral modes of the dynamics is 14.

5.1.2.3 The strongly nonlinear configuration

For the L96 model, we also provide a *strongly nonlinear* DA configuration, in which the only difference with the mildly nonlinear configuration defined in the previous paragraph is that the observation vector $\mathbf{y}$ is computed from the truth $\mathbf{x}^\mathrm{t}$ using

$$\mathbf{y}_k = \mathcal{H}\big(\mathbf{x}^\mathrm{t}_k\big) + \mathbf{e}^\mathrm{o}_k, \quad \mathbf{e}^\mathrm{o}_k \sim \mathcal{N}[\mathbf{0}, \mathbf{I}], \tag{5.4}$$

where the observation operator $\mathcal{H}$ is defined as

$$\mathcal{H} : \begin{cases} \mathbb{R}^{N_\mathrm{x}} & \to \mathbb{R}^{N_\mathrm{y}} = \mathbb{R}^{N_\mathrm{x}}, \\ \mathbf{x} & \mapsto \ln|\mathbf{x}|, \end{cases} \tag{5.5}$$

in which the logarithm and absolute values are applied element-wise.

This configuration is used, for example, by Poterjoy (2016) to asses the performances of his LPF algorithm. The nonlinearities are strong, and come from both the integration of the ODEs defined by equation (5.2), and the nonlinear observation operator. Figure 5.1 illustrates the L96 model in both configurations.

5.1.3 The barotropic vorticity model

5.1.3.1 The dynamical model

The barotropic vorticity (BV) model describes the evolution of the vorticity field ζ of a two-dimensional incompressible homogeneous fluid in the $x_1 - x_2$ plane. The time evolution of ζ is governed by the scalar equation

$$\frac{\partial \zeta}{\partial t} + \mathrm{J}(\psi, \zeta) = -\xi\zeta + \nu\Delta\zeta + F, \tag{5.6}$$

and ζ is related to the stream function ψ through

$$\Delta\psi = \zeta. \tag{5.7}$$

In equation (5.6), $\mathrm{J}(\psi, \zeta)$ is the advection of ζ by ψ, defined by

$$\mathrm{J}(\psi, \zeta) \triangleq \frac{\partial \psi}{\partial x_1}\frac{\partial \zeta}{\partial x_2} - \frac{\partial \psi}{\partial x_2}\frac{\partial \zeta}{\partial x_1}, \tag{5.8}$$

$\xi \in \mathbb{R}_+$ is the friction coefficient, $\nu \in \mathbb{R}_+$ is the diffusion coefficient, and F is the forcing term, which may depend on x_1, x_2 and on the time t. The system is characterised by homogeneous two-dimensional turbulence. The friction extracts energy at large scales, the diffusion dissipates vorticity at small scales and the forcing injects energy in the system. The number of degrees of freedom in this model can be roughly considered to be proportional to the number of vortices (Chris Snyder, personal communication, 2012).

The equations are solved with $P \times P$ grid points regularly distributed over the simulation domain $[0, L] \times [0, L]$ with doubly periodic boundary conditions. Our time integration method is based on a semi-Lagrangian solver with a constant integration time step δt.

1. At time t, solve equation (5.7) for the stream function ψ.

2. At time t, compute the advection velocity $\nabla\psi$ using second-order centred finite differences.

3. The advection of the vorticity field ζ during t and $t + \delta t$ is computed by applying a semi-Lagrangian method to the left-hand side of equation (5.6). The overall solver cannot be more accurate than first-order in time, since the value of the stream function ψ is not updated during this step. Therefore, our semi-Lagrangian solver uses the first-order forward Euler time integration method. The interpolation method used here is the cubic convolution interpolation algorithm, which is third-order accurate in space. During this step, the right-hand side of equation (5.6) is ignored.

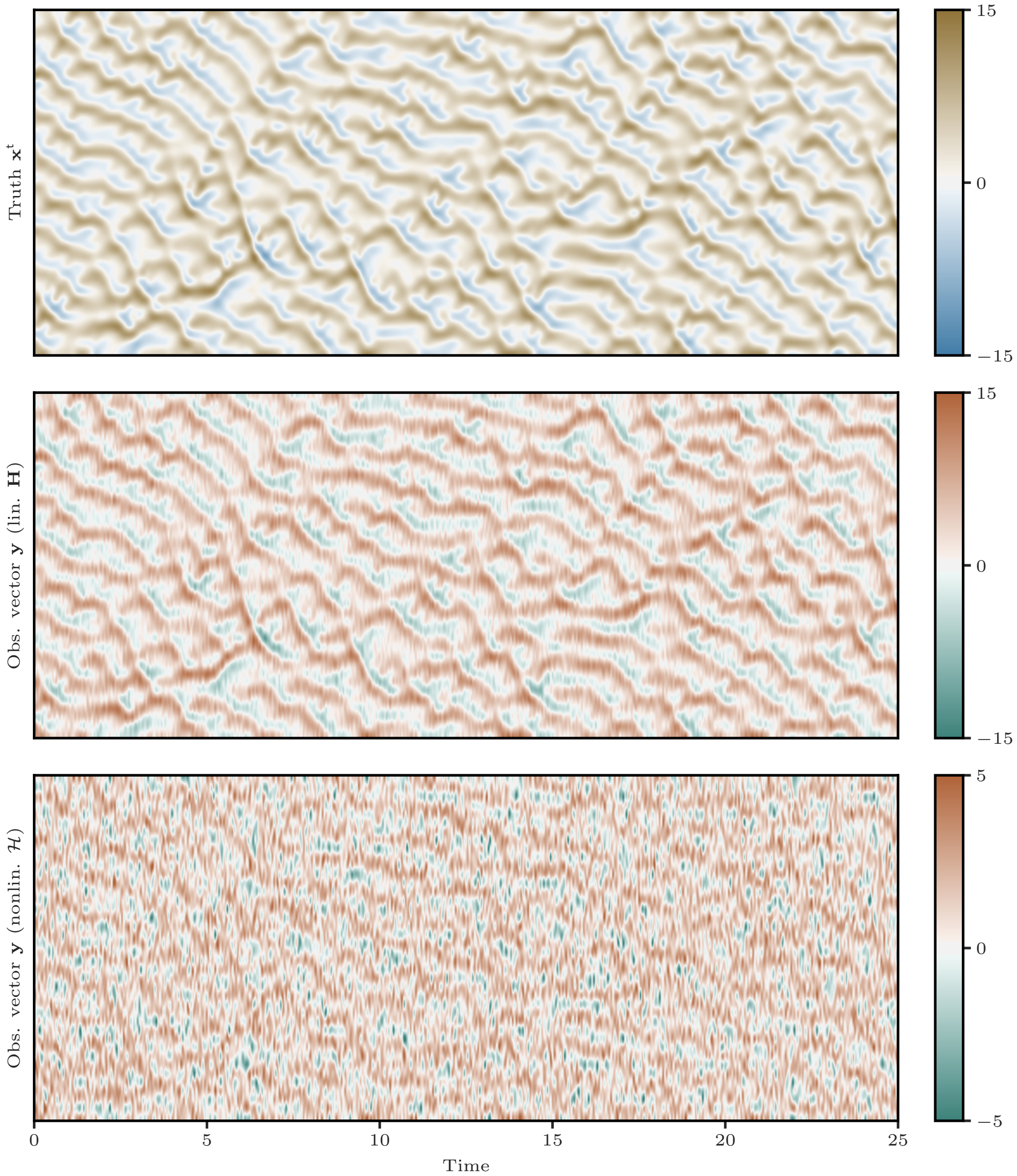

Figure 5.1: Illustration of the L96 model in the mildly and strongly nonlinear configurations. The top panel shows the trajectory of the $N_{\mathrm{x}} = 40$ variables of the truth $\mathbf{x}^{\mathrm{t}}$ during a time period of 25 units of time. The middle and bottom panel show the corresponding trajectory for the observation vector $\mathbf{y}$ in the mildly and strongly nonlinear configurations.

4. Integrate the vorticity field ζ from t and $t+\delta t$ by solving equation (5.6) with an implicit first-order time integration method in which the advection term is the one computed in the previous step.

For the numerical experiments of this chapter, the spatial discretisation is fine enough for the spatial interpolation error in the semi-Lagrangian method to be negligible compared to the time integration error. As a consequence, the overall integration method is first-order accurate in time. For the numerical experiments in this chapter, the integration time step δt is set to 0.1 unit of time. It was found to be empirically enough to ensure the stability of the integration method and it allows a fast computation of the trajectory.

5.1.3.2 The coarse-resolution configuration

For the BV model, we define a *coarse-resolution* configuration, with $P = 32$ grid points in each dimension. Empirically, this spatial discretisation is enough to allow a reasonable description of a few (typically five to ten) vortices inside the domain. The physical parameters are then chosen to ensure a proper time evolution of the vorticity field ζ. The simulation domain has a unit length, $L = 1$, the friction coefficient is $\xi = 1 \times 10^{-2}$, the diffusion coefficient is $\nu = 5 \times 10^{-5}$, and the deterministic forcing F is given by

$$F(x_1, x_2) = 0.25 \sin(4\pi x_1) \sin(4\pi x_2). \tag{5.9}$$

We have checked that the vorticity flow remains stationary over the total simulation time for all experiments presented in this chapter. Due to the forcing term F, the flow remains uniformly and stationarily turbulent during the whole simulation.

In the DA experiments, the control vector $\mathbf{x}$ is the vector containing the $P \times P$ values of the vorticity field ζ. The initial truth $\mathbf{x}^{\mathrm{t}}$ is the vorticity field ζ obtained after a run of a 100 time units starting from a random, spatially correlated field. The time interval between consecutive observations Δt, chosen to match approximately the model autocorrelation of 0.967 of the L96 model in the mildly nonlinear configuration, is set to 0.5 unit of time. Finally, the observation vector $\mathbf{y}$ is computed from the truth $\mathbf{x}^{\mathrm{t}}$ using

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}^{\mathrm{t}}_k + \mathbf{e}^{\mathrm{o}}_k, \quad \mathbf{e}^{\mathrm{o}}_k \sim \mathcal{N}\big[\mathbf{0}, r^2\mathbf{I}\big], \tag{5.10}$$

where the linear observation operator $\mathbf{H}$ corresponds to an interpolation using a regular, square mesh, with one observation site for every two grid points in each physical dimension:

$$\forall (q_1, q_2) \in (P/2\,{:}\,1)^2, \quad [\mathbf{H}\mathbf{x}]_{q_1,q_2} = [\mathbf{x}]_{2q_1-1,2q_2-1}. \tag{5.11}$$

The individual observation standard deviation is set to $r = 0.3$, about one tenth of the typical vorticity variability.

In this configuration, there are $N_{\mathrm{y}} = 256$ observations, regularly distributed in space, for a total of $N_{\mathrm{x}} = 1024$ state variables. Figure 5.2 illustrates the BV model in this configuration. Compared to other experiments with the BV model (*e.g.*, van Leeuwen and Ades 2013; Ades and van Leeuwen 2015; Browne 2016), the time interval between consecutive observations Δt is smaller, and the individual observation standard deviation r is larger, but the number of vortices is approximately the same, with much fewer details.

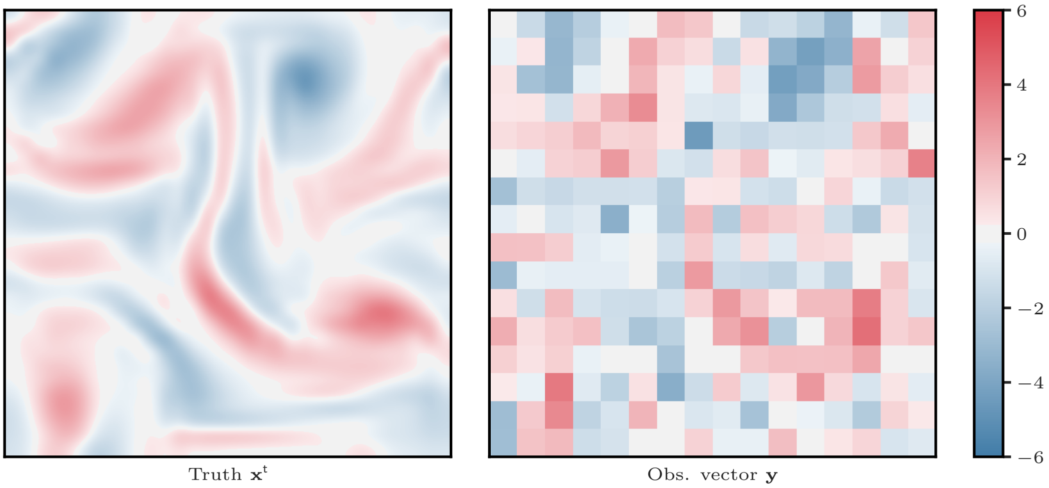

Figure 5.2: Illustration of the BV model in the coarse-resolution configuration. The left panel shows a snapshot of the $N_{\mathrm{x}} = 1024$ variables of the truth $\mathbf{x}^{\mathrm{t}}$ (with interpolation). The right panel shows the corresponding observation vector $\mathbf{y}$ with $N_{\mathrm{y}} = 256$ observations.

5.1.3.3 The high-resolution configuration

For the BV model, we define an *high-resolution* configuration, with $P = 256$ grid points in each dimension. Empirically, this spatial discretisation is enough to allow a reasonable description of a few dozen vortices inside the domain. Again, the physical parameters are chosen to ensure a proper time evolution of the vorticity field ζ. The simulation domain has a unit length, $L = 1$, the friction coefficient is $\xi = 5 \times 10^{-5}$, the diffusion coefficient is $\nu = 1 \times 10^{-6}$, and the deterministic forcing F is given by

$$F(x_1, x_2) = 0.75 \sin(12\pi x_1) \sin(12\pi x_2). \tag{5.12}$$

In the DA experiments, the control vector $\mathbf{x}$ is the vector containing the $P \times P$ values of the vorticity field ζ. The initial truth $\mathbf{x}^{\mathrm{t}}$ is the vorticity field ζ obtained after a run of a 100 time units starting from a random, spatially correlated field. The time interval between consecutive observations Δt is the same as in the coarse-resolution configuration, 0.5 unit of time. Finally, the observation vector $\mathbf{y}$ is computed from the truth $\mathbf{x}^{\mathrm{t}}$ using

$$\mathbf{y}_k = \mathbf{H}\mathbf{x}^{\mathrm{t}}_k + \mathbf{e}^{\mathrm{o}}_k, \quad \mathbf{e}^{\mathrm{o}}_k \sim \mathcal{N}\big[\mathbf{0}, r^2\mathbf{I}\big], \tag{5.13}$$

where the linear observation operator $\mathbf{H}$ corresponds to an interpolation using a regular, square mesh, with one observation site for every four grid points in each physical dimension:

$$\forall (q_1, q_2) \in (P/4\,{:}\,1)^2, \quad \big[\mathbf{H}\mathbf{x}\big]_{q_1,q_2} = \big[\mathbf{x}\big]_{4q_1-1,4q_2-1}. \tag{5.14}$$

The individual observation standard deviation is the same as in the coarse-resolution configur-

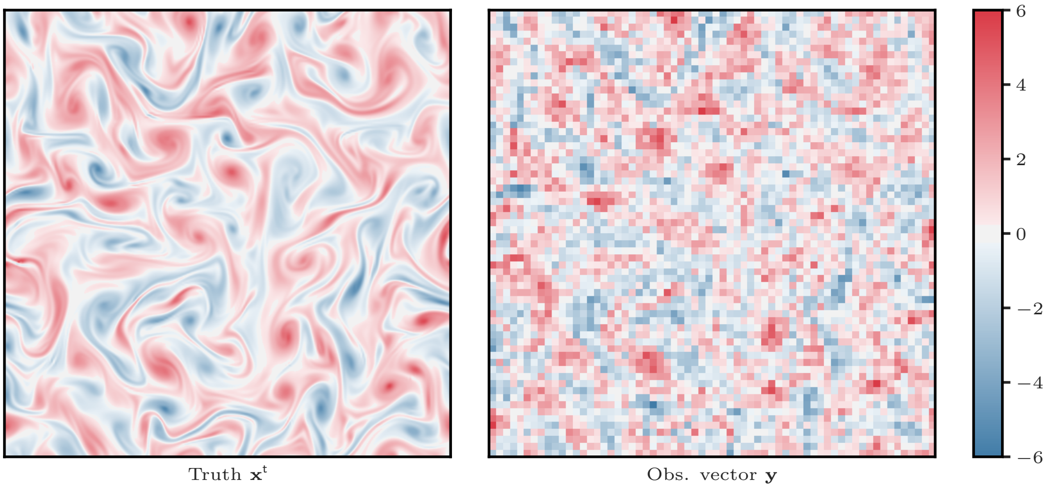

Figure 5.3: Illustration of the BV model in the high-resolution configuration. The top panel shows a snapshot of the $N_{\mathrm{x}} = 65\,536$ variables of the truth $\mathbf{x}^{\mathrm{t}}$. The bottom panel shows the corresponding observation vector $\mathbf{y}$ with $N_{\mathrm{y}} = 4096$ observations.

ation, $r = 0.3$. In this configuration, there are $N_{\mathrm{y}} = 4096$ observations, regularly distributed in space, for a total of $N_{\mathrm{x}} = 65\,536$ state variables. Figure 5.3 illustrates the BV model in this configuration.

5.2 Experiments with the L96 model

In this section, we illustrate the performance of several DA algorithms using twin experiments of the L96 model described in subsection 5.1.2. The algorithms are compared to the ETKF and the LETKF algorithms, algorithms 2.3 and 2.4. With the exception of subsection 5.2.8, we only consider the mildly nonlinear configuration of the L96 model, as described in paragraph 5.1.2.2.

We insist on the fact that, for this numerical experiments, as well as for all numerical experiments described in this chapter, the synthetic truth is computed without model error. This is usually a stringent constraint for the PF, for which accounting for the model error is a means for regularisation. But on the other hand, it allows for a fair comparison with the EnKF, and it avoids the issue of defining a realistic model error $\boldsymbol{e}^{\mathsf{m}}$.

In order to ensure the convergence of the statistical indicators, we use a spin-up period of $N_{\mathrm{s}} = 10^3$ assimilation cycles and a total simulation period of at least $N_{\mathrm{c}} \geq N_{\mathrm{s}} + 10^4$ assimilation cycles. For the localisation, is assumed that the N_{x} variables are positioned on an axis, with periodic boundary conditions consistent with the size of the system. The n-th grid point, corresponding to the n-th variable, has for coordinate n. Finally, for the LETKF algorithm, the localisation matrices $\boldsymbol{\rho}_{N_{\mathrm{x}}:1}$ are constructed using equation (2.57).

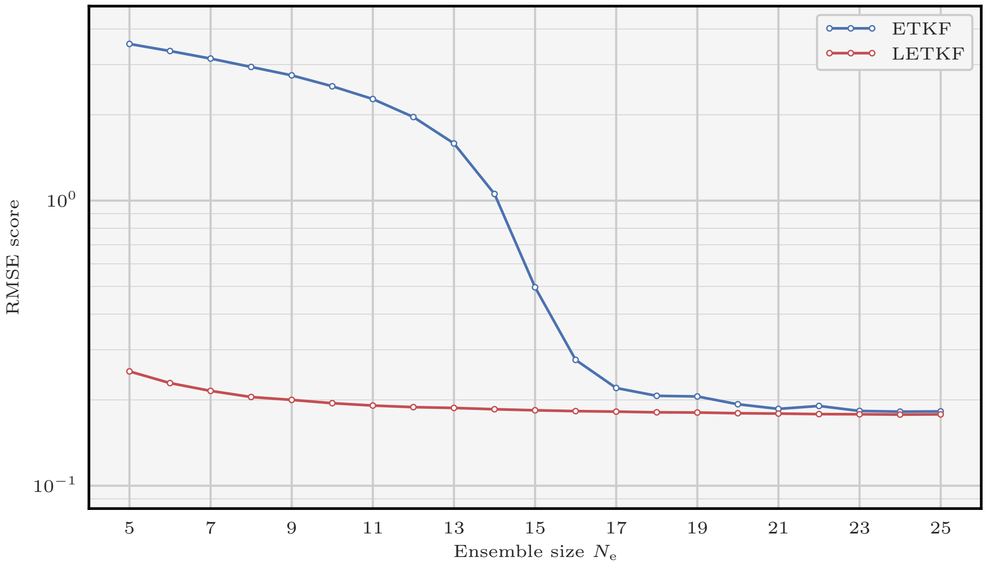

Figure 5.4: Evolution of the RMSE score as a function of the ensemble size N_e for the ETKF (in blue) and the LETKF (in red) algorithms. The DA system is the L96 model in the mildly nonlinear configuration.

5.2.1 Illustration of the EnKF

The mildly nonlinear configuration of the L96 model is widely used to illustrate the performance of the DA algorithms, and in particular of the EnKF. Figure 5.4 shows the evolution of the RMSE score as a function of the ensemble size N_e for the ETKF and the LETKF algorithms. As presented in subsection 2.5.2, in order to mitigate the sampling errors, multiplicative inflation is used after the analysis step with a fixed multiplicative inflation factor λ. For each value of the ensemble size N_e, the multiplicative inflation factor λ, as well as the localisation radius ℓ (only for the LETKF), are optimally tuned to yield the lowest RMSE score.

In this chapter, in most of the following figures related to the mildly nonlinear configuration, we draw a baseline at 0.2, roughly the RMSE score of the LETKF algorithm with $N_\mathrm{e} = 10$ ensemble members (even though slightly lower RMSE scores can be achieved with larger ensembles).

5.2.2 Illustration of the global PF

The application of the PF to this DA system without model error leads to a fast and irremediable collapse. As described in paragraph 3.3.2.2, the sample impoverishment phenomenon, which is known to cause the collapse, can be counteracted with regularisation. The regularisation step can be implemented in two different ways (Musso et al. 2001): as pre-regularisation, which is equivalent to use an additional model error $\boldsymbol{e}^\mathsf{m}$, and as post-regularisation, for example with equation (3.56).

Algorithm 5.1: Full assimilation cycle for the regularised SIR algorithm in a DA system with a deterministic model $\mathcal{M}$.

```
Input: $\mathbf{E}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\mathcal{M}\ [t_k \rightarrow t_{k+1}]$, $\pi^{\mathrm{o}}\ [t_{k+1}]$, $q$, $s$
Sampling
    $\mathbf{E}^{\mathsf{m}} \overset{\text{iid}}{\sim} \mathcal{N}\big[\mathbf{0}, q^2\mathbf{I}\big]$
    $\mathbf{E} \leftarrow \mathcal{M}(\mathbf{E}) + \mathbf{E}^{\mathsf{m}}$        // pre-regularisation
Importance
    $\mathbf{w} \leftarrow \pi^{\mathrm{o}}(\mathbf{y}|\mathbf{E})$
Resampling
    $\mathbf{E}^{\mathsf{r}} \leftarrow \mathtt{Resampling}(\mathbf{w}, \mathbf{E})$
    $\mathbf{E}^{\mathsf{m}} \overset{\text{iid}}{\sim} \mathcal{N}\big[\mathbf{0}, s^2\mathbf{I}\big]$
    $\mathbf{E} \leftarrow \mathbf{E}^{\mathsf{r}} + \mathbf{E}^{\mathsf{m}}$        // post-regularisation
Output: $\mathbf{E}\ [t_{k+1}]$
```

5.2.2.1 Implementation of pre- and post-regularisation

For the numerical experiments with the L96 model, the regularisation step is implemented as follows:

- for pre-regularisation, the additional model error $\boldsymbol{e}^{\mathsf{m}}$ has distribution $\mathcal{N}\big[\mathbf{0}, q^2\mathbf{I}\big]$;
- for post-regularisation, the additional jitter added after the resampling step is drawn from the distribution $\mathcal{N}\big[\mathbf{0}, s^2\mathbf{I}\big]$;

where the pre- and post-regularisation standard deviations q and s are two parameters to be determined. Algorithm 5.1 describes a full assimilation cycle for the resulting regularised SIR algorithm. The regularised SIR algorithm is an approximation of the original (non-regularised) SIR algorithm, algorithm 3.2. In the limit $q \rightarrow 0$ and $s \rightarrow 0$, both algorithms are equivalent.

Figure 5.5 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for the regularised SIR algorithm using only pre-regularisation (*i.e.*, with $s = 0$), only post-regularisation (*i.e.*, with $q = 0$), or both pre- and post-regularisation. In either case, the resampling step is performed using the systematic resampling algorithm (algorithm 3.4) instead of the multinomial resampling algorithm (algorithm 3.3). For each value of the ensemble size N_{e}, the pre- and post-regularisation standard deviations q and s are optimally tuned to yield the lowest RMSE score.

The SIR algorithm requires more than $N_{\mathrm{e}} = 128$ particles to give, on average, more informations than the observations.[1] With $N_{\mathrm{e}} = 8196$ particles, the RMSE score of the SIR

[1] The RMSE score between the observations $\mathbf{y}$ and the truth $\mathbf{x}^{\mathrm{t}}$ has an expected value around 0.98 in this

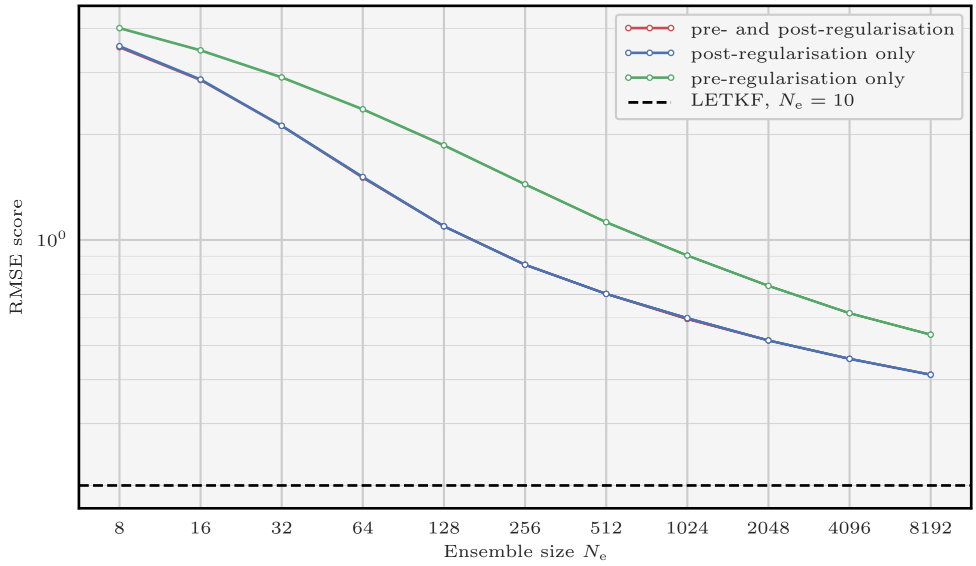

Figure 5.5: Evolution of the RMSE score as a function of the ensemble size N_e for the regularised SIR algorithm, using only post-regularisation (*i.e.*, with $q = 0$, in blue), using only pre-regularisation (*i.e.*, with $s = 0$, in green), or using both pre- and post-regularisation (in red). For comparison, the RMSE score of the LETKF algorithm with $N_\mathrm{e} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration. Visually, the RMSE scores obtained using only post-regularisation can hardly be distinguished from the RMSE scores obtained using both pre- and post-regularisation.

algorithm is not even close to the RMSE score of the ETKF algorithm. In this experiment, post-regularisation is much more efficient that pre-regularisation. This can be explained as follows. With post-regularisation, the jitter is added after the importance and resampling step. As a consequence, the jitter is integrated by the dynamical model $\mathcal{M}$ before the next importance and resampling steps, which smoothes potential unphysical discontinuities. By contrast, with pre-regularisation, the additional model noise is added just before the importance and resampling step, which means that potential unphysical discontinuities are included in the posterior ensemble $\mathbf{E}$.

Furthermore, when post-regularisation is optimally tuned, the additional tuning of pre-regularisation has almost no effect on the RMSE score. The same tendency is observed in all numerical experiments tested in this chapter. Therefore, from now on, we only implement post-regularisation in the PF and LPF algorithms.

5.2.2.2 The regularised SIR and ETPF algorithms

Without model error, all proposal distributions in the PF are equivalent to the standard proposal distribution $\nu^{\mathsf{q}} = \nu^{\mathsf{m}}$. As a consequence, the difference between PF algorithms only come from the resampling step. In this paragraph, we illustrate the benefits of replacing the resampling step by a linear transformation step, as suggested in paragraph 3.3.3.1.

Figure 5.6 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for the regularised SIR and ETPF algorithms. The regularised SIR algorithm is algorithm 5.1, in which the pre-regularisation standard deviation q is set to zero, and the resampling step is performed with the systematic resampling algorithm. The regularised ETPF algorithm is algorithm 3.5, in which a post-regularisation step is included as described in the previous paragraph. For each value of the ensemble size N_{e}, the post-regularisation standard deviation s is optimally tuned to yield the lowest RMSE score. These results confirm the advantages of the ETPF algorithm over the SIR algorithm listed in paragraph 3.3.3.1.

5.2.2.3 Colourising the regularisation

For simplicity, in this paragraph the time index k is systematically dropped.

With post-regularisation as introduced in paragraph 5.2.2.1, the additional jitter is a white noise. In realistic models, different state variables may take their values in disjoint intervals, which makes white jittering methods inadequate. Therefore, it is a common technique in ensemble DA to scale the jitter with statistical properties of the ensemble $\mathbf{E}$. In the PF, practitioners often *colourise* the Gaussian jitter with the empirical covariances of the ensemble $\mathbf{E}$ as described by Musso et al. (2001). Since the jitter is added after the resampling step, it is scaled with the weighted ensemble $(\mathbf{w}, \mathbf{E})$ before resampling in order to mitigate the effect of the sampling noise. In this case, the covariance matrix of the Gaussian jitter is the posterior sample covariance matrix $\bar{\mathbf{P}}$, whose n-th row, m-th column element is given by

$$\left[\bar{\mathbf{P}}\right]_{n,m} = \frac{h}{1 - \bar{\mathbf{w}}^{\mathsf{T}}\bar{\mathbf{w}}} \sum_{i=1}^{N_{\mathrm{e}}} \bar{w}(i) \left[x_n(i) - \bar{x}_n\right] \left[x_m(i) - \bar{x}_m\right]. \tag{5.15}$$

configuration with $N_{\mathrm{x}} = 40$ variables.

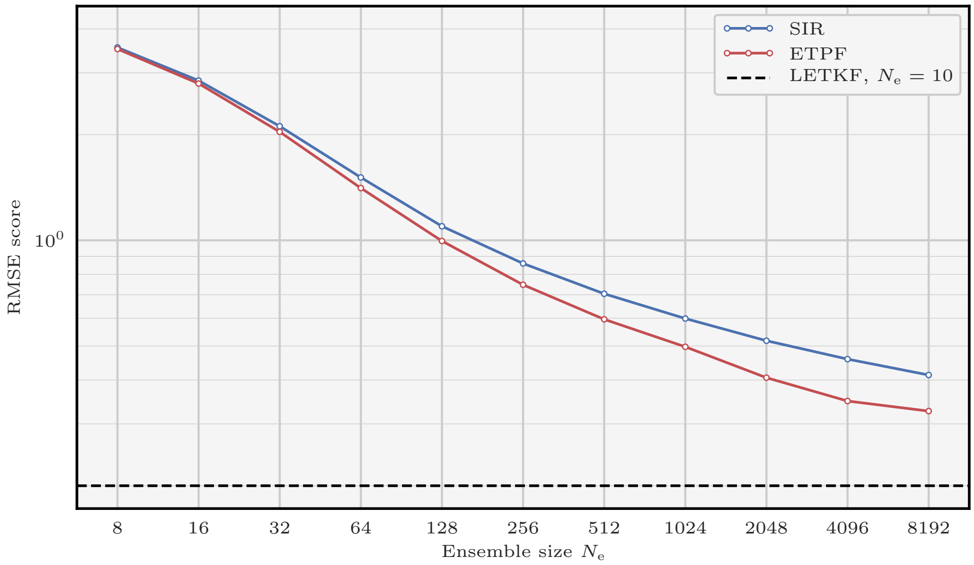

Figure 5.6: Evolution of the RMSE score as a function of the ensemble size N_e for the regularised SIR (in blue) and ETPF (in red) algorithms. For comparison, the RMSE score of the LETKF algorithm with $N_\text{e} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

In this equation, the **bandwidth** $h \in \mathbb{R}_+$ is a parameter to be determined, $x_n(i)$ is the n-th variable of the i-th particle $\mathbf{x}(i)$, and $\bar{x}_n$ is the n-th variable of the posterior ensemble mean $\bar{\mathbf{x}} = \mathbf{E}\bar{\mathbf{w}}$.

In practice, we define the normalised anomaly matrix $\mathbf{X}$ as the $N_\text{x} \times N_\text{e}$ matrix whose n-th row, i-th column element is

$$[\mathbf{X}]_{n,i} = \left(\frac{h\bar{w}(i)}{1 - \bar{\mathbf{w}}^\mathsf{T}\bar{\mathbf{w}}} \right)^{1/2} \big[x_n(i) - \bar{x}_n\big], \tag{5.16}$$

and the regularisation jitter is added to the ensemble $\mathbf{E}$ as

$$\mathbf{E} \leftarrow \mathbf{E} + \mathbf{XZ}, \tag{5.17}$$

where $\mathbf{Z}$ is an iid sample (of size N_e) from the normal distribution $\mathcal{N}[\mathbf{0}, \mathbf{I}]$ in the ensemble space $\mathbb{R}^{N_\text{e}}$, in such a way that $\mathbf{XZ}$ is an iid sample from the normal distribution $\mathcal{N}\big[\mathbf{0}, \bar{\mathbf{P}}\big]$.

Algorithm 5.2 describes a full assimilation cycle for the resulting regularised SIR algorithm, with colourised post-regularisation. Algorithm 5.2 is another approximation of the original SIR algorithm, algorithm 3.2. Again, both algorithms are equivalent in the limit $h \rightarrow 0$.

Figure 5.7 shows the evolution of the RMSE score as a function of the ensemble size N_e for the regularised SIR algorithm with white and with colourised post-regularisation. It turns out that, in the mildly nonlinear configuration of the L96 model, using colourised post-regularisation (*i.e.*, with algorithm 5.2) yields much higher RMSE scores than using

Algorithm 5.2: Full assimilation cycle for the regularised SIR algorithm, with colourised post-regularisation.

```
Input: $\mathbf{E}\ [t_k]$, $\mathbf{y}\ [t_{k+1}]$
Parameters: $\mathcal{M}\ [t_k \rightarrow t_{k+1}]$, $\pi^{\mathrm{o}}\ [t_{k+1}]$
Sampling
    $\mathbf{E} \leftarrow \mathcal{M}(\mathbf{E})$                    // no pre-regularisation
Importance
    $\mathbf{w} \leftarrow \pi^{\mathrm{o}}(\mathbf{y}|\mathbf{E})$
Resampling
    $\mathbf{X} \leftarrow$ equation (5.16)
    $\mathbf{E}^{\mathrm{r}} \leftarrow \mathtt{Resampling}(\mathbf{w}, \mathbf{E})$
    $\mathbf{Z} \overset{\mathrm{iid}}{\sim} \mathcal{N}[\mathbf{0}, \mathbf{I}]$                    // in $\mathbb{R}^{N_{\mathrm{e}}}$
    $\mathbf{E} \leftarrow \mathbf{E}^{\mathrm{r}} + \mathbf{X}\mathbf{Z}$                    // colourised post-regularisation
Output: $\mathbf{E}\ [t_{k+1}]$
```

white post-regularisation (*i.e.*, with algorithm 5.1) unless the number of particles N_{e} is very high ($N_{\mathrm{e}} \geq 2048$). The relative success of the white post-regularisation method, compared to the colourised post-regularisation method can be explained by two factors. First, in the L96 model, the $N_{\mathrm{x}} = 40$ variables are statistically homogeneous with short-range correlations. Second, in this configurations where the error distributions are close to Gaussian, the posterior sample covariance matrix $\bar{\mathbf{P}}$ is a poor approximation of the (exact) posterior covariance matrix $\mathbf{P}$, unless the number of particles N_{e} is very high.

5.2.3 Towards an implementation of LPF algorithms

5.2.3.1 The standard LPF–X algorithm

In this subsection, we explain step-by-step the implementation of an LPF algorithm by taking the example of the standard LPF–X algorithm, defined in this thesis as the LPF–X algorithm (algorithm 4.1) with the following characteristics.

- Grid points are gathered into N_{b} local blocks of $N_{\mathrm{x}}/N_{\mathrm{b}}$ connected grid points.
- The local importance weight vectors $\mathbf{w}_{N_{\mathrm{b}}:1}$ are computed using the Gaussian formulation, with equation (4.25).
- The local resampling is performed independently for each local block using the adjustment-minimising systematic resampling algorithm (algorithm 3.4 with the modification described in paragraph 4.2.3.3).

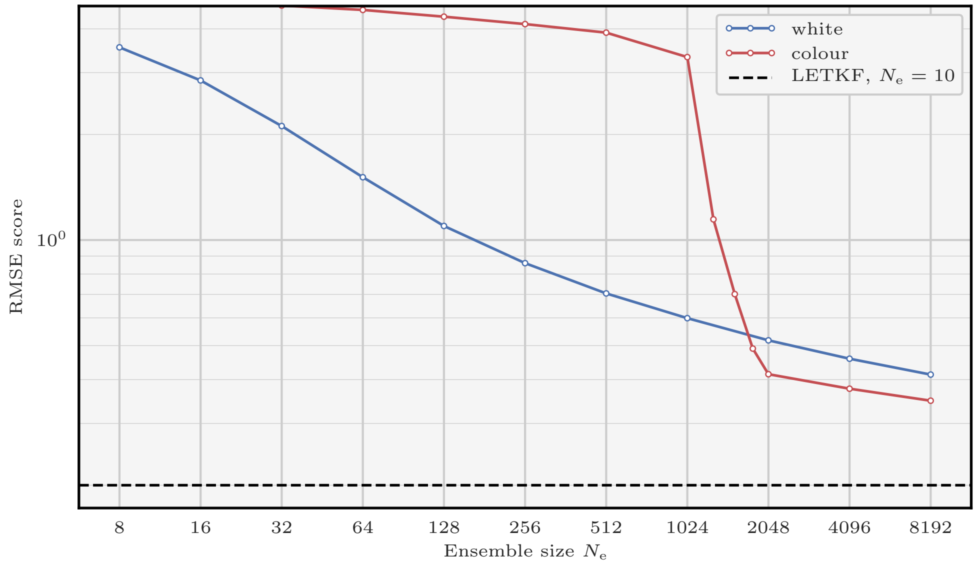

Figure 5.7: Evolution of the RMSE score as a function of the ensemble size N_e for the regularised SIR with white (in blue) and with colourised (in red) post-regularisation. For comparison, the RMSE score of the LETKF algorithm with $N_e = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

Algorithm 5.3: Full assimilation cycle for the standard LPF–X algorithm in a DA system with a deterministic model $\mathcal{M}$.

```
Input: $\mathbf{E}$ $[t_k]$, $\mathbf{y}$ $[t_{k+1}]$
Parameters: $\mathcal{M}$ $[t_k \rightarrow t_{k+1}]$, $\pi^{\text{o}}$ $[t_{k+1}]$, $N_{\text{b}}$, $\ell$, $s$
Sampling
    $\mathbf{E} \leftarrow \mathcal{M}(\mathbf{E})$
Importance
    for $b = 1$ to $N_{\text{b}}$ do
        $\mathbf{w}_b \leftarrow$ equation (4.25)
    end
Resampling
    for $b = 1$ to $N_{\text{b}}$ do
        $\mathbf{E}^{\mathsf{r}}_{|b} \leftarrow \texttt{Resampling}\big(\mathbf{w}_b, \mathbf{E}_{|b}\big)$
    end
    $\mathbf{E}^{\mathsf{r}} \leftarrow \texttt{Assembling}\big(\mathbf{E}^{\mathsf{r}}_{|N_{\text{b}}:1}\big)$
    $\mathbf{E}^{\mathsf{m}} \overset{\text{iid}}{\sim} \mathcal{N}\big[\mathbf{0}, s^2\mathbf{I}\big]$
    $\mathbf{E} \leftarrow \mathbf{E}^{\mathsf{r}} + \mathbf{E}^{\mathsf{m}}$
Output: $\mathbf{E}$ $[t_{k+1}]$
```

- Regularisation jitter is added after each assimilation cycle using a white post-regularisation step, as described in paragraph 5.2.2.1.

This is summarised by algorithm 5.3. Besides the ensemble size N_{e}, the standard LPF–X algorithm has three parameters: the number of local blocks N_{b}, the localisation radius ℓ, and the post-regularisation standard deviation s.

5.2.3.2 Tuning the localisation and the post-regularisation

We first check that localisation is working in this configuration, by testing the standard LPF–X algorithm with $N_{\text{b}} = 40$ local blocks. We take $N_{\text{e}} = 10$ particles, and several values for the post-regularisation standard deviation s are used. The evolution of the RMSE score as a function of the localisation radius ℓ is shown in figure 5.8. With localisation, the LPF yields an RMSE score around 0.45 in a regime where the regularised SIR algorithm is degenerate. The compromise between bias (small values of ℓ, too much information is ignored, or there is too much spatial variation in the local importance weight vectors $\mathbf{w}_{N_{\text{b}}:1}$) and variance (large values of ℓ, the $\mathbf{w}_{N_{\text{b}}:1}$ are degenerate) reaches an optimum around $\ell = 3$ grid points.

As expected, the local domains are quite small (5 observation sites) in order to efficiently counteract the curse of dimensionality.

To evaluate the efficiency of the post-regularisation, we experiment with the standard LPF–X algorithm using $N_{\mathrm{e}} = 10$ particles, $N_{\mathrm{b}} = 40$ local blocks, and several values of the localisation radius ℓ. The evolution of the RMSE score as a function of s is shown in figure 5.8. The compromise between perfect model (small values of s, the ensemble $\mathbf{E}$ collapses because of the sample impoverishment phenomenon) and perturbed model (high values of s, too much noise is added) reaches an optimum around $s = 0.26$.

Similar behaviours are observed for all LPF algorithms tested in this chapter. From now on, ℓ and s are systematically tuned to yield the lowest RMSE score.

5.2.3.3 Choosing the number of local blocks

To illustrate the influence of the size of the local blocks, we compare the RMSE scores obtained by the standard LPF–X algorithm using various number of local blocks N_{b}. The evolution of the RMSE score as a function of the ensemble size N_{e} is shown in figure 5.9. For small ensembles, using larger local blocks is inefficient, because of the need for degrees of freedom to counteract the curse of dimensionality. Only very large ensembles benefit from using large local blocks as a consequence of the reduction of proportion of block boundaries (which are potential spots for unphysical discontinuities).

5.2.3.4 Formulation of the local importance weight vectors

To illustrate the influence of the formulation of the local importance weight vector $\mathbf{w}_{N_{\mathrm{b}}:1}$, the standard LPF–X algorithm is compared with a variant thereof, in which the $\mathbf{w}_{N_{\mathrm{b}}:1}$ are computed using the generic formulation, with equation (4.22). Figure 5.10 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for both algorithms. Using the Gaussian formulation for the $\mathbf{w}_{N_{\mathrm{b}}:1}$ always yields lower RMSE scores. This is probably a consequence of the fact that, in this configuration, the nonlinearities are weak and the error distributions are close to Gaussian.

5.2.3.5 Refinements of the resampling methods

In this paragraph, the refinements of the resampling algorithms proposed in paragraph 4.2.3.3 are tested. To do this, the standard LPF–X algorithm is compared with the following two variants.

- In the first variant, the same random number is used for the resampling of each local block with the adjustment-minimising systematic resampling algorithm.
- In the second variant, the systematic resampling algorithm (algorithm 3.4) is used as is, in other words without the adjustment-minimising property.

Figure 5.11 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for all three algorithms. The second variant, the only algorithm here which does not use an adjustment-minimising resampling algorithm for the local updates, yields higher RMSE scores. This shows that the adjustment-minimising property is indeed an efficient way of

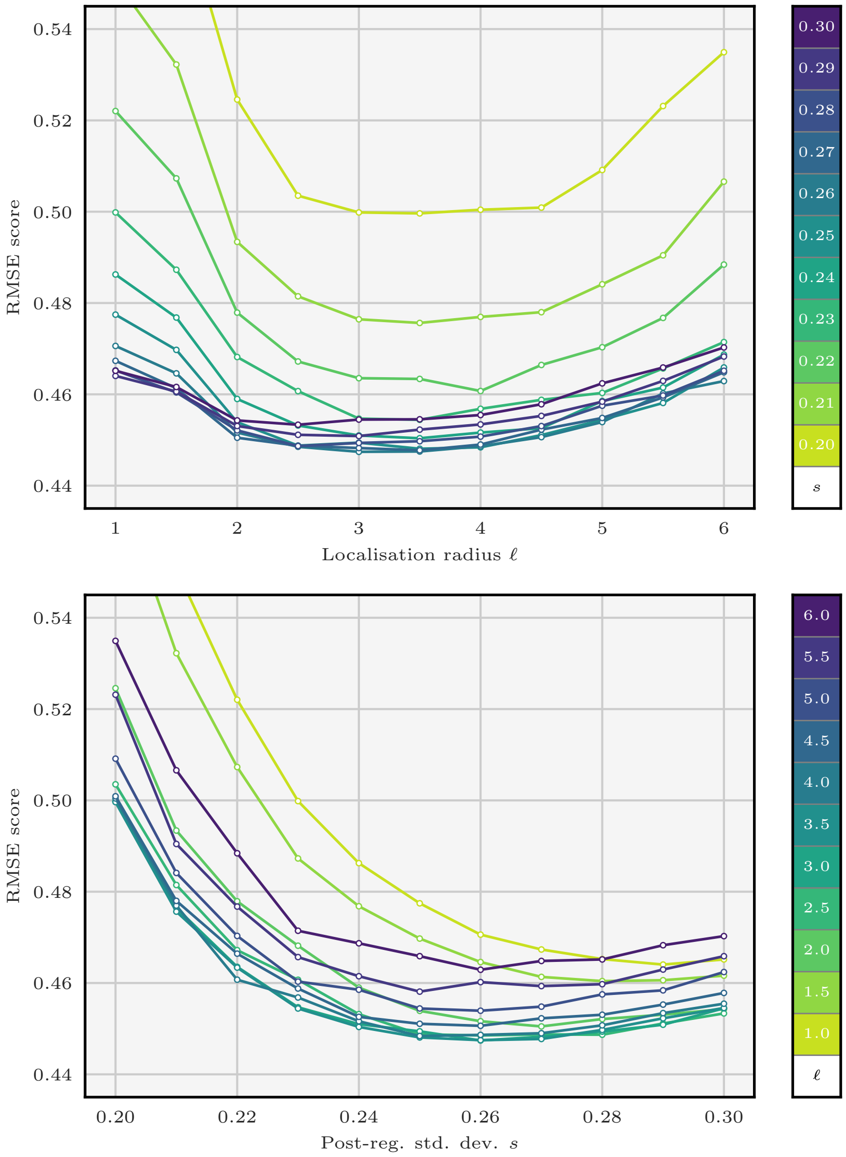

Figure 5.8: Evolution of the RMSE score of the standard LPF–X algorithm as a function of the localisation radius ℓ (in number of grid points) for several values of the post-regularisation standard deviation s (top panel), and as a function of the post-regularisation standard deviation s for several values of the localisation radius ℓ (bottom panel). In both cases, the algorithm uses $N_{\mathrm{e}} = 10$ particles and $N_{\mathrm{b}} = 40$ local blocks. The DA system is the L96 model in the mildly nonlinear configuration.

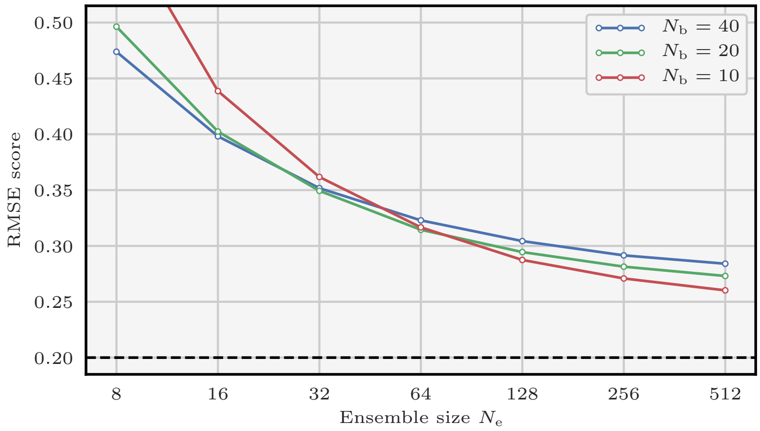

Figure 5.9: Evolution of the RMSE score as a function of the ensemble size N_{e} for the standard LPF–X algorithm using $N_{\mathrm{b}} = 40$ (in blue), 20 (in green), and 10 (in red) local blocks with a size of 1, 2, and 4 grid points, respectively. For comparison, the RMSE score of the LETKF algorithm with $N_{\mathrm{e}} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

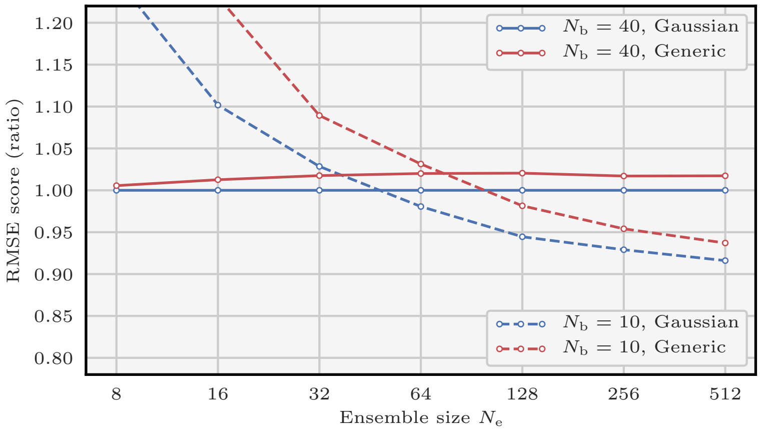

Figure 5.10: Evolution of the RMSE score as a function of the ensemble size N_{e} for the standard LPF–X algorithm (in blue) and its variant using the generic formulation of the local importance weight vectors $\mathbf{w}_{N_{\mathrm{b}}:1}$ (in red), with $N_{\mathrm{b}} = 40$ (continuous lines), and 10 (dashed lines) local blocks. The DA system is the L96 model in the mildly nonlinear configuration. In order to emphasise the differences, the RMSE scores are divided by the RMSE score of the standard LPF–X algorithm using $N_{\mathrm{b}} = 40$ local blocks (continuous blue line).

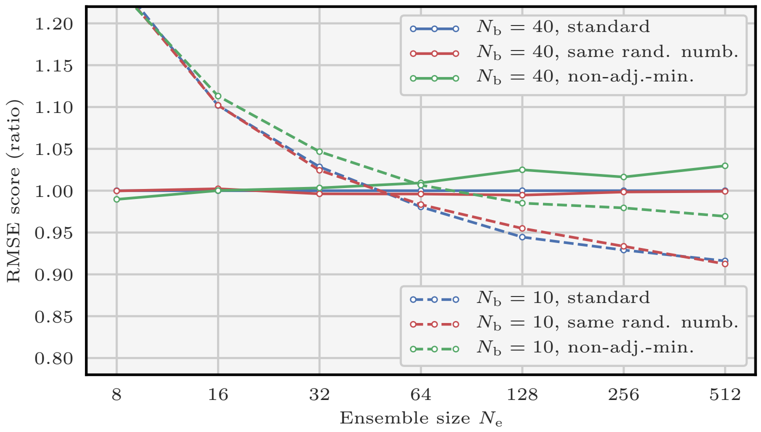

Figure 5.11: Evolution of the RMSE score as a function of the ensemble size N_e for the standard LPF–X algorithm (in blue), for its variant using the same random numbers for the resampling of each local block (in red), and for its variant using the non-adjustment-minimising systematic resampling algorithm (in green). In all three cases, the algorithms use $N_b = 40$ (continuous lines) or $N_b = 10$ (dashed lines) local blocks. The DA system is the L96 model in the mildly nonlinear configuration. In order to emphasise the differences, the RMSE scores are divided by the RMSE score of the standard LPF–X algorithm using $N_b = 40$ local blocks (continuous blue line).

reducing the number of unphysical discontinuities introduced during the resampling step. By constrast, using the same random number for the resampling of each local block does not yield significantly lower RMSE scores: this method is insufficient to reduce the number of unphysical discontinuities introduced when assembling the locally updated particles. This is probably a consequence of the fact that the systematic resampling algorithm only uses one random number to compute the resampling map ψ. It also suggests that the specific realisation of this random number has a weak influence on the long-term statistical properties.

5.2.3.6 Colourisation of the regularisation

In this paragraph, the potential benefits of using colourisation in the post-regularisation step, as presented in paragraph 5.2.2.3, are investigated. For simplicity, the time index k is systematically dropped.

As is, the method presented in paragraph 5.2.2.3 cannot be directly applied. Indeed with the LPF–X algorithms, there is one importance weight vector $\mathbf{w}$ per local block, and therefore the normalised anomaly matrix $\mathbf{X}$ cannot be computed using equation (5.16). Two different methods can be used to circumvent this obstacle.

A first approach could be to scale the regularisation with the locally resampled ensemble $\mathbf{E}^{\mathrm{r}}$, because after the local resampling, the $\mathbf{w}_{N_b:1}$ have been reset. This is the approach followed, *e.g.*, by Reich (2013) and Chustagulprom et al. (2016) under the name *particle rejuvenation*. However, preliminary experiments (not illustrated here) have shown that this approach systematically yields higher RMSE scores than using a white post-regularisation step. This can be potentially explained by two factors. First, the resampling step introduces sampling noise, which is included in the resulting anomaly matrix **X**. Second, the fact that the resampling step is performed independently for each local block perturbs the propagation of multivariate properties (such as the covariance) over different local blocks.

In a second approach, the anomaly matrix **X** is defined as the $N_\mathrm{x} \times N_\mathrm{e}$ matrix whose n-th row, i-th column element is

$$[\mathbf{X}]_{n,i} = \left(\frac{h\bar{w}_n(i)}{1-\bar{\mathbf{w}}_n^\mathsf{T}\bar{\mathbf{w}}_n}\right)^{1/2}\big[x_n(i)-\bar{x}_n\big], \tag{5.18}$$

where $\mathbf{w}_n$ is the local importance weight vector corresponding to the local block in which the n-th variable is located. In this case, the covariance matrix $\bar{\mathbf{P}}$ of the Gaussian jitter has n-th row, m-th column element given by

$$\big[\bar{\mathbf{P}}\big]_{n,m} = h\sum_{i=1}^{N_\mathrm{e}}\left[\frac{\bar{w}_n(i)\,\bar{w}_m(i)}{(1-\bar{\mathbf{w}}_n^\mathsf{T}\bar{\mathbf{w}}_n)(1-\bar{\mathbf{w}}_m^\mathsf{T}\bar{\mathbf{w}}_m)}\right]^{1/2}\big[x_n(i)-\bar{x}_n\big]\big[x_m(i)-\bar{x}_m\big], \tag{5.19}$$

which is a generalisation of equation (5.15). This method can also be seen as a generalisation of the adaptive inflation used by Penny and Miyoshi (2016), in which only the diagonal of the anomaly matrix **X** is computed and in which the post-regularisation bandwidth h is set to 1. In all tested cases, our method systematically yields lower RMSE scores than the method of Penny and Miyoshi (2016), which is most probably due to the tuning of the post-regularisation bandwidth h.

From a practical point of view, preliminary experiments (not illustrated here) have shown that the evolution of the RMSE score as a function of the post-regularisation bandwidth h, for LPF algorithms using a colourised post-regularisation step, is similar to the evolution of the RMSE score as a function of the post-regularisation standard deviation s, for LPF algorithms using a white post-regularisation step, as described in paragraph 5.2.3.2.

Finally, in order to illustrate the benefits of using a colourised post-regularisation step, the standard LPF–X algorithm is compared with a variant thereof, in which the post-regularisation step is colourised. Figure 5.12 shows the evolution of the RMSE score as a function of the ensemble size N_e for both algorithms, in which the tuning of s is replaced by the tuning of h for the variant using colourised post-regularisation. For small ensembles, using a colourised post-regularisation step yields higher RMSE scores, whereas it shows slightly better RMSE scores for large ensembles. Depending on the number of local blocks N_b, the transition between both regimes happens when the ensemble size N_e is between 32 to 64 particles.

From now on, when using a colourised post-regularisation step, the tuning of of the post-regularisation standard deviation s, mentioned in paragraph 5.2.3.2, is systematically replaced by the tuning of the post-regularisation bandwidth h.

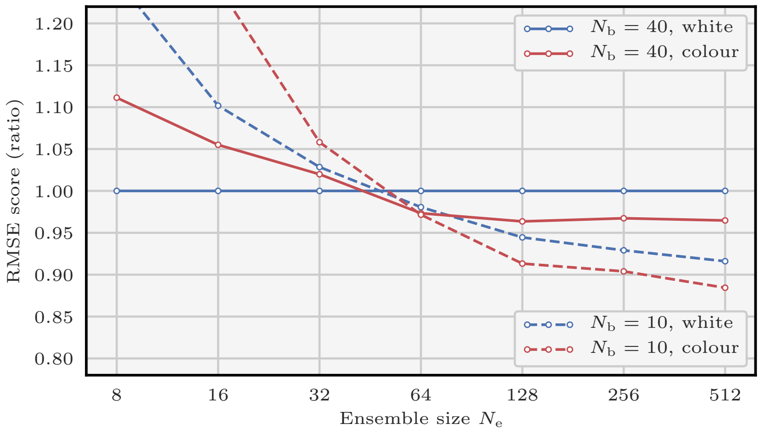

Figure 5.12: Evolution of the RMSE score as a function of the ensemble size N_e for the standard LPF–X algorithm (in blue) and its variant using colourised post-regularisation (in red), with $N_b = 40$ (continuous lines), and 10 (dashed lines) local blocks. The DA system is the L96 model in the mildly nonlinear configuration. In order to emphasise the differences, the RMSE scores are divided by the RMSE score of the standard LPF–X algorithm using $N_b = 40$ local blocks (continuous blue line).

5.2.4 Illustration of the LPF–X algorithms

In the following paragraphs, we illustrate the performances of several LPF–X algorithms. In order to distinguish between the different algorithmic variants, the legends of the figures follow the same simple convention, explained in table 5.1. Following this convention, the label of the standard LPF–X algorithm is `sys/w`.

5.2.4.1 The LPF–X with the smoothing-by-weights step

In this paragraph, we examine the potential benefit of adding a smoothing-by-weights step, presented in paragraph 4.2.3.1. Alongside the smoothing-by-weights come two additional parameters: the smoothing strength α^s and the smoothing radius ℓ^s. We first investigate the influence of theses parameters. To do this, we compare the standard LPF–X algorithm with a variant thereof, in which a smoothing-by-weights step is added after the local resampling step.

Figure 5.13 shows the evolution of the RMSE score of the LPF–X algorithm with smoothing-by-weights as a function of ℓ^s, for several values of α^s. In these experiments, the ensemble size is $N_e = 16$, and the number of local blocks is $N_b = 40$. For a fixed smoothing strength $\ell^s > 0$, starting from $\ell^s = 1$ grid point (no smoothing), the RMSE score decreases when ℓ^s increases. It reaches a minimum, and then increases again. In this case, the optimal ℓ^s lies between 5 and 6 grid points when $\alpha^s = 1$, with a corresponding optimal localisation radius ℓ between 2 and 3 grid points and optimal post-regularisation standard deviation s around 0.45. For comparison, the optimal tuning parameters for the standard LPF–X algorithm (without smoothing-by-weights) are approximately $\ell = 4.5$ grid points and $s = 0.2$.

Table 5.1: Nomenclature convention for the LPF–X algorithms. The local updates are performed either with the adjustment-minimising systematic resampling algorithm (`sys`), with the optimal ensemble coupling (`oec`), or with the anamorphosis (`ana`). The smoothing-by-weights step is either enabled (`smo`) or disabled, and the post-regularisation step is either white (`w`) or colourised (`c`).

Label	Local update method	Smoothing-by-weights	Post-regularisation
`sys/w`	sys. resampling	–	white
`sys/c`	sys. resampling	–	colourised
`sys/smo/w`	sys. resampling	✓	white
`sys/smo/c`	sys. resampling	✓	colourised
`oec/w`	opt. ens. coupling	–	white
`oec/c`	opt. ens. coupling	–	colourised
`ana/w`	anamorphosis	–	white
`ana/c`	anamorphosis	–	colourised

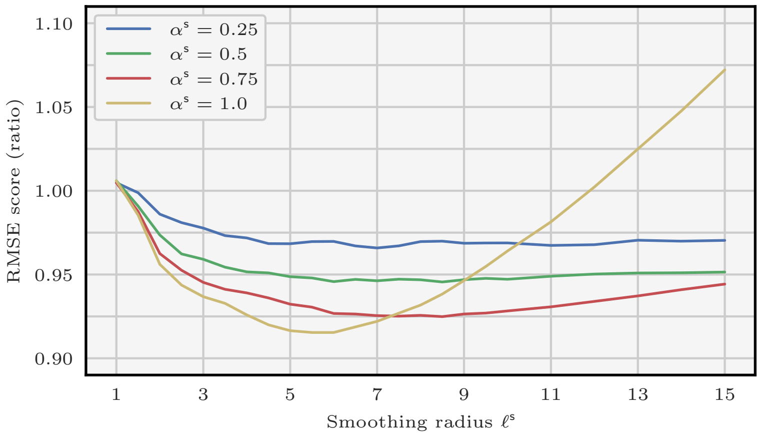

Figure 5.13: Evolution of the RMSE score as a function of the smoothing radius ℓ^s for the LPF–X algorithm with systematic resampling and with smoothing-by-weights. The algorithm uses $N_\mathrm{e} = 16$ particles and $N_\mathrm{b} = 40$ local blocks, and several values of the smoothing strength α^s are tested. The DA system is the L96 model in the mildly nonlinear configuration. In order to emphasise the differences, the RMSE scores are divided by the RMSE score of the standard LPF–X algorithm using $N_\mathrm{e} = 16$ particles and $N_\mathrm{b} = 40$ local blocks.

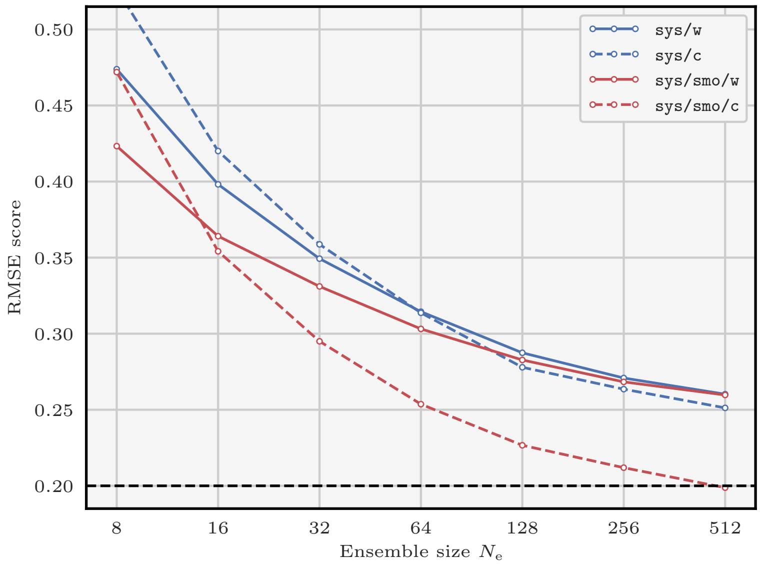

Figure 5.14: Evolution of the RMSE score as a function of the ensemble size N_e for the LPF–X algorithm with systematic resampling. The smoothing-by-weights step is enabled (in red) or not (in blue) and the post-regularisation step is white (continuous lines) or colourised (dashed lines). The DA system is the L96 model in the mildly nonlinear configuration.

Based on extensive tests of LPF–X algorithms using smoothing-by-weights with an ensemble size N_e ranging from 8 to 128 particles (not illustrated here), we draw the following conclusions. In general, using $\alpha^s = 1$ is optimal, or at least only slightly suboptimal. Optimal values ℓ and s are larger with smoothing-by-weights than without. Finally, optimal values for ℓ and ℓ^s are not related and must be tuned separately.

In order to further illustrate the influence of the smoothing-by-weights, the standard LPF–X algorithm is compared with the following three variants:

- in the first variant (`sys/c`), the post-regularisation is colourised, as described in paragraph 5.2.3.6;
- in the second variant (`sys/smo/w`), a smoothing-by-weights step is added after the local resampling step;
- in the third variant (`sys/smo/c`), a smoothing-by-weights step is added after the local resampling step and the post-regularisation is colourised.

Figure 5.14 shows the evolution of the RMSE score as a function of the ensemble size N_e for all four algorithms. For the second, and third variant, α^s is set to 1 and ℓ^s is optimally tuned to yield the lowest RMSE score. For each value of the ensemble size N_e, each experiment is performed with $N_b = 40$, 20, and 10 local blocks, and the lowest RMSE score is kept.

The second variant (which uses smoothing-by-weights and white post-regularisation) systematically yields lower RMSE scores than the standard LPF–X algorithm (without smoothing-by-weights and with white post-regularisation). However, as the ensemble size N_e grows, the gain in RMSE score becomes very small. With $N_\mathrm{e} = 512$ particles, there is almost no difference between both scores. In this case, the optimal ℓ^s is around 5 grid points, much smaller than the optimal ℓ, around 15 grid points, in such a way that the smoothing-by-weights step does not modify much the analysis ensemble $\mathbf{E}^\mathsf{a}$. The third variant (which uses smoothing-by-weights and colourised post-regularisation) yields lower RMSE scores than the standard LPF–X algorithm as well. Yet, in this case, the gain in RMSE score is still significant for large ensembles, and with $N_\mathrm{e} = 512$ particles, the RMSE score is even comparable to that of the EnKF.

From these results, we conclude that the smoothing-by-weights is an efficient way of mitigating the unphysical discontinuities which are introduced when assembling the locally updated particles, especially when combined with colourised post-regularisation.

5.2.4.2 The LPF–X with optimal ensemble coupling

In this paragraph, we evaluate the efficiency of replacing the local resampling step by a linear transformation step based on the optimal ensemble coupling, presented in paragraph 4.2.3.4. For each local block, the LET matrix $\mathbf{T}_\mathrm{e}$ is computed by solving a minimisation problem which can be seen as a particular case of the minimum-cost flow problem. We have chosen to compute its numerical solution using the network simplex algorithm implemented in the graph library LEMON (Dezső et al. 2011). This method is characterised by an additional parameter: the distance radius ℓ^d, which is used to define the local cost coefficients $\mathcal{C}_b\big(\mathbf{E}^\mathsf{f}\big)$ with equation (4.32).

The influence of the number of local blocks N_b and of the distance radius ℓ^d has been first investigated with extensive tests of LPF–X algorithms using optimal ensemble coupling, for an ensemble size N_e ranging from 8 to 128 particles (not illustrated here). The following conclusions are drawn. Optimal values for ℓ^d are much smaller than those of ℓ, and are even smaller than 2 grid points most of the time. Using $\ell^\mathsf{d} = 1$ grid point yields RMSE scores which are only very slightly suboptimal. Furthermore, all other things being equal, using $N_\mathrm{b} = 20$ local blocks systematically yields higher RMSE scores than using $N_\mathrm{b} = 40$ local blocks. Finally, the optimal ensemble coupling can be combined with an additional smoothing-by-weights step. The resulting algorithm is significantly more costly. For small ensembles (typically $N_\mathrm{e} \leq 32$ particles), the RMSE scores are barely smaller with smoothing-by-weights than without. For larger ensembles, we could not find a set of parameters for which using smoothing-by-weights yields lower RMSE scores.

In order to illustrate the influence of using the optimal ensemble coupling, the standard LPF–X algorithm is compared with the following three variants.

- In the first variant (`sys/c`), the post-regularisation step is colourised.
- In the second variant (`oec/w`), the local resampling step is replaced by a linear transformation step using the optimal ensemble coupling.
- In the third variant (`oec/c`), the local resampling step is replaced by a linear transformation step using the optimal ensemble coupling and the post-regularisation step is colourised.

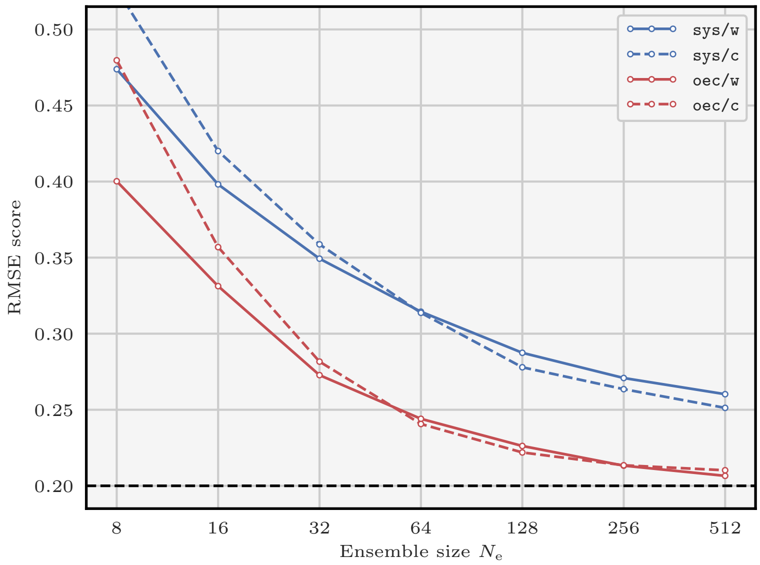

Figure 5.15: Evolution of the RMSE score as a function of the ensemble size N_e for the LPF–X algorithm with systematic resampling (in blue) and with optimal ensemble coupling (in red). The post-regularisation step is white (continuous lines) or colourised (dashed lines). For comparison, the RMSE score of the LETKF algorithm with $N_\mathrm{e} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

Figure 5.15 shows the evolution of the RMSE score as a function of the ensemble size N_e for all four algorithms. For the second, and third variant, the distance radius ℓ^d is set to 1. For each value of the ensemble size N_e, each experiment is performed with $N_\mathrm{b} = 40$, 20, and 10 local blocks, and the lowest RMSE score is kept. For the second and third variant, the lowest RMSE score is always the one obtained with $N_\mathrm{b} = 40$ local blocks.

Using optimal ensemble coupling for the local updates systematically yields significantly lower RMSE scores than using systematic resampling. By contrast with the results of the previous paragraph, using colourised post-regularisation does not improve the RMSE scores for very large ensembles. From the fact that neither the use of larger local blocks nor the use of smoothing-by-weights further improves the RMSE scores with optimal ensemble coupling, we conclude that this local update method is indeed an efficient way of mitigating the unphysical discontinuities inherent to assembling the locally updated particles.

5.2.4.3 The LPF–X with anamorphosis

In this paragraph, we evaluate the efficiency of replacing the local resampling step by a transport step based on the anamorphosis, presented in paragraph 4.2.3.5. For each state variable $n \in (N_\mathrm{x} : 1)$, the transport map $\mathcal{T}_n$ is computed using the cdf of the n-th regularised marginal empirical forecast and analysis densities $\bar{\pi}_n^\mathsf{f}$ and $\bar{\pi}_n^\mathsf{a}$, as defined by equations (4.35) and (4.36). The regularisation kernel $\mathcal{K}$ is chosen as the Student's t-distribution with two degrees of freedom. This method is characterised by two additional parameters: the forecast and analysis regularisation bandwidths h^{f} and h^{a}, which are used to define $\bar{\pi}^{\mathsf{f}}_{N_{\mathrm{x}}:1}$ and $\bar{\pi}^{\mathsf{a}}_{N_{\mathrm{x}}:1}$ with equations (4.35) and (4.36).

The influence of h^{f} and h^{a} has been first investigated with extensive tests of LPF–X algorithms using anamorphosis, for an ensemble size N_{e} ranging from 8 to 128 particles (not illustrated here). The following conclusions are drawn. For small ensembles (typically $N_{\mathrm{e}} \leq 16$ particles), optimal values for h^{f} and h^{a} are found to lie between 2 and 3, the RMSE score obtained with $h^{\mathsf{f}} = h^{\mathsf{a}} = 1$ being very slightly suboptimal. For larger ensembles, we did not find any significant difference in RMSE score between $h^{\mathsf{f}} = h^{\mathsf{a}} = 1$ and larger values for h^{f} and h^{a}. Finally, the anamorphosis can be combined with an additional smoothing-by-weights step, and the conclusion are similar as for the optimal ensemble coupling. The resulting algorithm is significantly more costly. For small ensembles, the RMSE scores are barely smaller with smoothing-by-weights than without, and for larger ensembles, we could not find a set of parameters for which using smoothing-by-weights yields lower RMSE scores.

In order to illustrate the influence of using the anamorphosis, the standard LPF–X algorithm is compared with the following three variants:

- In the first variant (`sys/c`), the post-regularisation step is colourised.
- In the second variant (`ana/w`), the local resampling step is replaced by a transport step using the anamorphosis.
- In the third variant (`ana/c`), the local resampling step is replaced by a transport step using the anamorphosis and the post-regularisation step is colourised.

Figure 5.16 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for all four algorithms. For the second, and third variant, h^{f} and h^{a} are set to 1, and $N_{\mathrm{b}} = N_{\mathrm{x}} = 40$ local blocks are used, because the anamorphosis is only defined in one dimension. For the standard LPF–X algorithm and its first variant, for each value of the ensemble size N_{e}, each experiment is performed with $N_{\mathrm{b}} = 40$, 20, and 10 local blocks, and the lowest RMSE score is kept.

Using anamorphosis for the local updates yields RMSE scores even lower than when using optimal ensemble coupling. However in this case, using colourised post-regularisation systematically yields significantly higher RMSE scores than using white post-regularisation. This is probably a consequence of the fact that some colourised regularisation is already introduced in the local update through the kernel representation of $\bar{\pi}^{\mathsf{f}}_{N_{\mathrm{x}}:1}$ and $\bar{\pi}^{\mathsf{a}}_{N_{\mathrm{x}}:1}$. From the fact that the use of smoothing-by-weights does not further improve the RMSE scores with anamorphosis, and from the significantly lower RMSE scores obtained when using anamorphosis, we conclude that the local update based on anamorphosis is, as well as the local update based on optimal ensemble coupling, an efficient way of mitigating the unphysical discontinuities inherent to assembling the locally updated particles.

5.2.5 Illustration of the LPF–Y algorithms

In the following paragraphs, we illustrate the performances of several LPF–Y algorithms. In order to distinguish between the different algorithmic variants, the legends of the figures follow the same simple convention, explained in table 5.2.

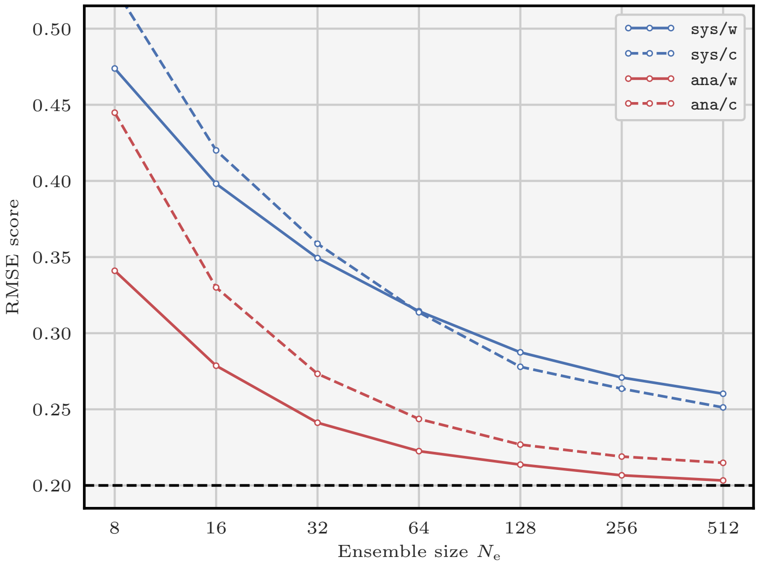

Figure 5.16: Evolution of the RMSE score as a function of the ensemble size N_e for the LPF–X algorithm with systematic resampling (in blue) and with anamorphosis (in red). The post-regularisation step is white (continuous lines) or colourised (dashed lines). For comparison, the RMSE score of the LETKF algorithm with $N_e = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

Table 5.2: Nomenclature convention for the LPF–Y algorithms. The propagation method is either the hybrid propagation method (`hyb`) or the second-order propagation method (`so`). In the latter case, the local updates on the U regions are performed either using the adjustment-minimising systematic resampling algorithm (`sys`), with the optimal ensemble coupling (`oec`), or with the anamorphosis (`ana`). Finally, the post-regularisation step is either white (`w`) or colourised (`c`).

Label	Local update method	Propagation method	Post-regularisation
`sys/hyb/w`	sys. resampling	hybrid	white
`sys/hyb/c`	sys. resampling	hybrid	colourised
`sys/so/w`	sys. resampling	second-order	white
`sys/so/c`	sys. resampling	second-order	colourised
`oec/so/w`	opt. ens. coupling	second-order	white
`oec/so/c`	opt. ens. coupling	second-order	colourised
`ana/so/w`	anamorphosis	second-order	white
`ana/so/c`	anamorphosis	second-order	colourised

5.2.5.1 The LPF–Y with the hybrid propagation method

In this paragraph, we illustrate the performance of the LPF–Y algorithm using the hybrid propagation method, described in subsection 4.3.2. In order to avoid a fast collapse of the algorithm, a post-regularisation step is added after each assimilation cycle (*i.e.*, after the N_{y} local sequential updates). As for the LPF–X algorithms, the additional post-regularisation step can be either white (as presented in paragraph 5.2.2.1) or a colourised (as presented in paragraph 5.2.3.6). In the latter case, the local importance weight vectors $\mathbf{w}_{N_{\mathrm{x}}:1}$ required for the computation of the anomaly matrix $\mathbf{X}$ with equation (5.18) are computed using the Gaussian formulation, with equation (4.25).

In the original LPF algorithm of Poterjoy (2016), the collapse of the algorithm is mitigated by using a weight inflation method. Based on extensive tests of LPF–Y algorithms using the hybrid propagation method, with an ensemble size N_{e} ranging from 8 to 128 particles (not illustrated here), we conclude that using the weight inflation method systematically yields higher RMSE scores than using the additional post-regularisation step. Therefore, the weight inflation method is not included in our implementation of the LPF–Y algorithms using the hybrid propagation method.

Figure 5.17 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for the LPF–Y algorithm using the hybrid propagation method. For each value of the ensemble size N_{e}, the localisation radius ℓ, used to compute the prior and posterior update weights ω^{f} and ω^{a} with equations (4.60) to (4.62), the post-regularisation standard deviation s (for white post-regularisation), and the post-regularisation bandwidth h (for colourised post-regularisation) are optimally tuned to yield the lowest RMSE score. The RMSE scores obtained with this method are comparable to those obtained with the standard LPF–X algorithm. Again, using colourised post-regularisation improves the RMSE scores for large ensembles.

5.2.5.2 The LPF–Y with the second-order propagation method

In this paragraph, we illustrate the performance of the LPF–Y algorithm using the second-order propagation method, described in subsection 4.3.3. Again, to avoid a fast collapse of the algorithm, a post-regularisation step is added after each assimilation cycle, exactly like in the previous paragraph. As suggested in paragraph 4.3.3.4, the resampling step on the U region can be replaced by a linear transformation step, using the optimal ensemble coupling, or by a transport step, using the anamorphosis.

Figure 5.17 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for the LPF–Y algorithm using the second-order propagation method. For each value of the ensemble size N_{e}, the localisation radius ℓ, used to compute the localised prior sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$ with equations (4.63) and (4.64), the post-regularisation standard deviation s (for white post-regularisation), and the post-regularisation bandwidth h (for colourised post-regularisation) are optimally tuned to yield the lowest RMSE score. Following the conclusions from paragraphs 5.2.4.2 and 5.2.4.3, when using the optimal ensemble coupling, the distance radius ℓ^{d} is set to 1 grid point, and when using the anamorphosis, the forecast and analysis regularisation bandwidths h^{f} and h^{a} are set to 1.

As expected, when using the second-order propagation method, the resulting LPF–Y

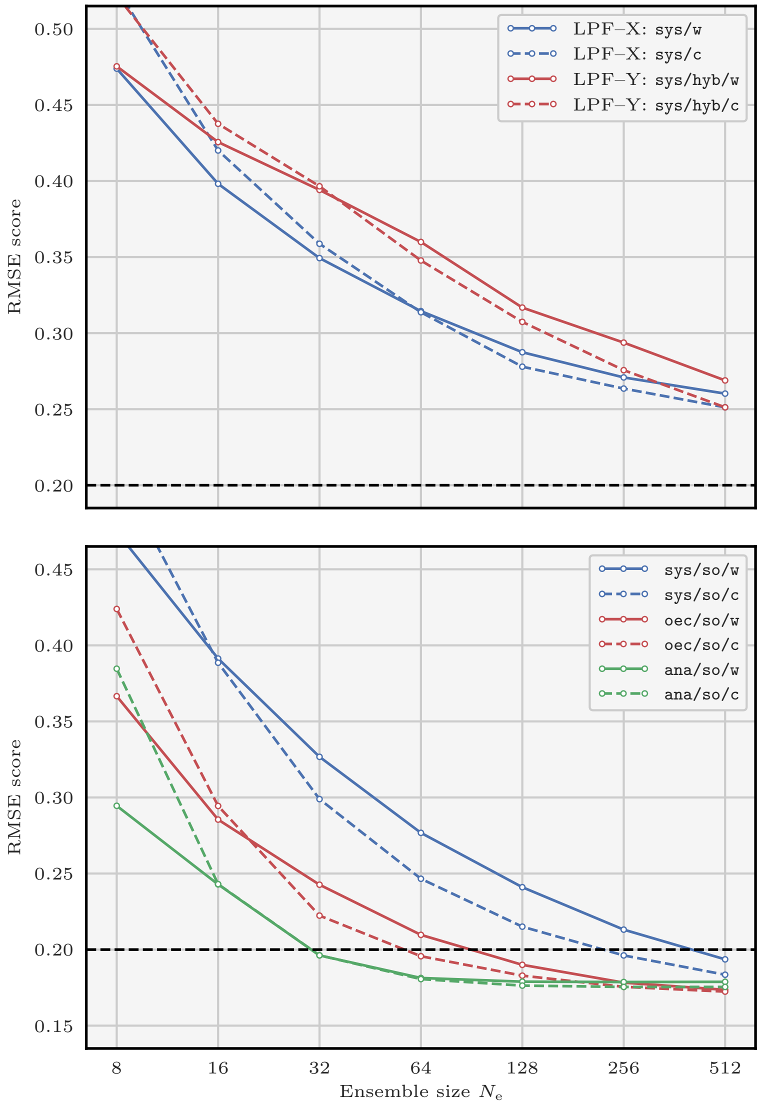

Figure 5.17: Evolution of the RMSE score as a function of the ensemble size N_e for the LPF–X algorithm with systematic resampling (top panel, in blue), for the LPF–Y algorithm using the hybrid propagation method (top panel, in red), for the LPF–Y algorithm using the second-order propagation method with systematic resampling (bottom panel, in blue), with optimal ensemble coupling (bottom panel, in red), or with anamorphosis (bottom panel, in green). The post-regularisation step is white (continuous lines) or colourised (dashed lines). For comparison, the RMSE score of the LETKF algorithm with $N_e = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

algorithms are less sensitive to the curse of dimensionality: compared to all other LPF algorithms, the RMSE scores are lower, the optimal values for ℓ are larger, and the optimal values for s are smaller. Some conclusions are similar as for the LPF–X algorithms. Using colourised post-regularisation yields lower RMSE scores for large ensembles only when combined with systematic resampling. Using a local update method based on the optimal transport theory (either optimal ensemble coupling or anamorphosis) results in important gain in RMSE scores, as a consequence of the minimisation of the updates in the U regions which need to be propagated to the V regions. With a reasonable number of particles (*e.g.*, $N_\mathrm{e} = 64$ when using the anamorphosis), the RMSE scores are significantly lower than those obtained with the reference EnKF algorithm.

5.2.6 Summary for the LPF algorithms

To summarise, figure 5.18 shows the evolution of the RMSE score as a function of the ensemble size N_e for a selection of LPF–X and LPF–Y algorithms, whose implementation is described in subsections 5.2.4 and 5.2.5.

- With small ensembles (typically $N_\mathrm{e} \leq 64$ particles), using optimal transport for the local updates yields the best scores.
- With large ensembles (typically $N_\mathrm{e} \geq 128$ particles), combining smoothing-by-weights and colourised post-regularisation in the LPF–X algorithms yields equally good RMSE scores as using optimal transport for the local updates.
- With very large ensembles ($N_\mathrm{e} = 512$), the best RMSE scores of the LPF–X algorithms become comparable to those of the EnKF.
- With only $N_\mathrm{e} \geq 64$ particles, the best RMSE scores for the LPF–Y algorithms are significantly lower than those of the EnKF.

5.2.7 Rank histograms for the LPF algorithms

As a complement to the RMSE score, rank histograms are computed for the L96 model. For this experiment, four algorithms are selected:

- the ETKF algorithm;
- the LPF–X algorithm with systematic resampling and with white post-regularisation (`sys/w`);
- the LPF–X algorithm with anamorphosis and with white post-regularisation (`ana/w`);
- the LPF–Y algorithm using the second-order propagation method, in which the local updates are performed with the anamorphosis, and with white post-regularisation (`ana/so/w`);

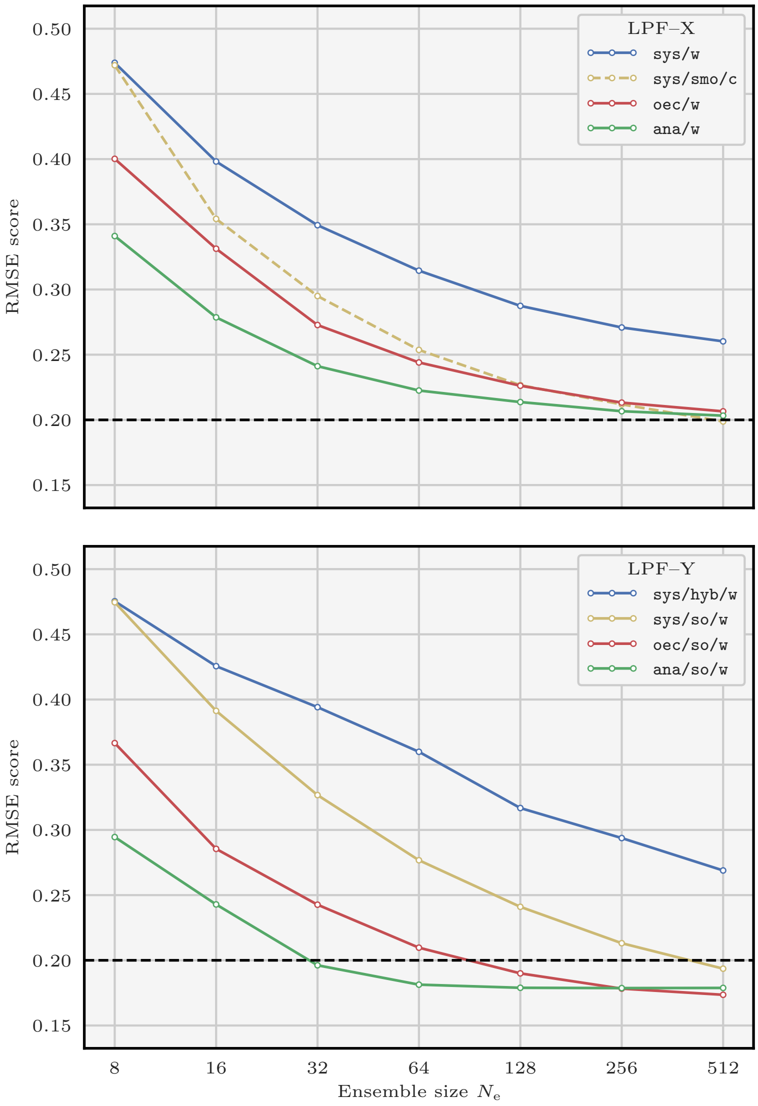

Figure 5.18: Evolution of the RMSE score as a function of the ensemble size N_e for the main LPF–X (top panel) and LPF–Y (bottom panel) algorithms. For comparison, the RMSE score of the LETKF algorithm with $N_\mathrm{e} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the mildly nonlinear configuration.

Table 5.3: Characteristics of the DA assimilation algorithms whose rank histograms are shown in figure 5.19. The first column indicates the corresponding label in figure 5.19. The localisation radius ℓ is given in number of grid points. An asterisk in the last column indicates that the algorithm parameters have been tuned to yield the lowest RMSE score.

Label	N_e	ℓ	s	Other param.	RMSE
LPF–X: `sys/w`	128	8	10.0×10^{-1}	$N_\mathrm{b} = 10$	0.289*
LPF–X: `sys/w/•`	128	5	8.0×10^{-2}	$N_\mathrm{b} = 40$	0.500
LPF–X: `ana/w`	128	20	4.5×10^{-1}	$h^\mathsf{f} = h^\mathsf{a} = 1$	0.215*
LPF–X: `ana/w/•`	128	10	3.0×10^{-1}	$h^\mathsf{f} = h^\mathsf{a} = 1$	0.228
LPF–Y: `ana/so/w`	128	80	1.0×10^{-2}	$h^\mathsf{f} = h^\mathsf{a} = 1$	0.180*
ETKF	20	∞	–	$\lambda = 1.02$	0.188*

The algorithmic parameters are reported in table 5.3. The rank histograms are obtained separately for each state variable $n \in (N_\mathrm{x} : 1)$ by computing the rank of the n-th variable of the truth $\mathbf{x}^\mathrm{t}$ in the unperturbed analysis ensemble $\mathbf{E}^\mathsf{a}$ (*i.e.*, the analysis ensemble $\mathbf{E}^\mathsf{a}$ before the post-regularisation step for the LPF algorithms). The mean histograms (averaged over all state variables) are reported in figure 5.19.

The histogram of the ETKF algorithm (top left, in blue) is quite flat in the middle, and its edges reflect a small overdispersion. The histogram of the tuned LPF–X algorithm with systematic resampling (top right, in red) is characterised by a large hump, showing that the analysis ensemble $\mathbf{E}^\mathsf{a}$ is overdispersive. At the same time, the high frequencies at the edges show that the algorithm yields a poor representation of the distribution tails (a very common trait in most PF algorithms). The overdispersion of the $\mathbf{E}^\mathsf{a}$ is a consequence of the fact that the parameters have been tuned to yield the lowest RMSE score, regardless of the flatness of the rank histogram. With a different set of parameter, the non-tuned LPF–X algorithm with systematic resampling (bottom right, in cyan) yields a much flatter histogram. In this case, the post-regularisation standard deviation s is lower, which explains the fact that $\mathbf{E}^\mathsf{a}$ is less overdispersive, and the localisation radius ℓ is smaller, in order to avoid the collapse of the algorithm. Of course, the RMSE score for the non-tuned LPF–X algorithm with systematic resampling is higher than for its tuned version. Similar conclusions can be found with the histograms of the tuned and non-tuned LPF–X algorithm with anamorphosis (central panels, in green and in yellow). In this case the histograms are significantly flatter than with the LPF–X algorithm with systematic resampling. Finally, the histogram of the (tuned) LPF–Y algorithm with anamorphosis (bottom left, in purple) is remarkably flat.

In summary, the rank histograms of the LPF algorithms are in general rather flat. The ensemble is more or less overdispersive, as a consequence of the use of post-regularisation, necessary to avoid the collapse of the algorithm. As most PF algorithms, the LPF algorithms yield a poor representation of the distribution tails.

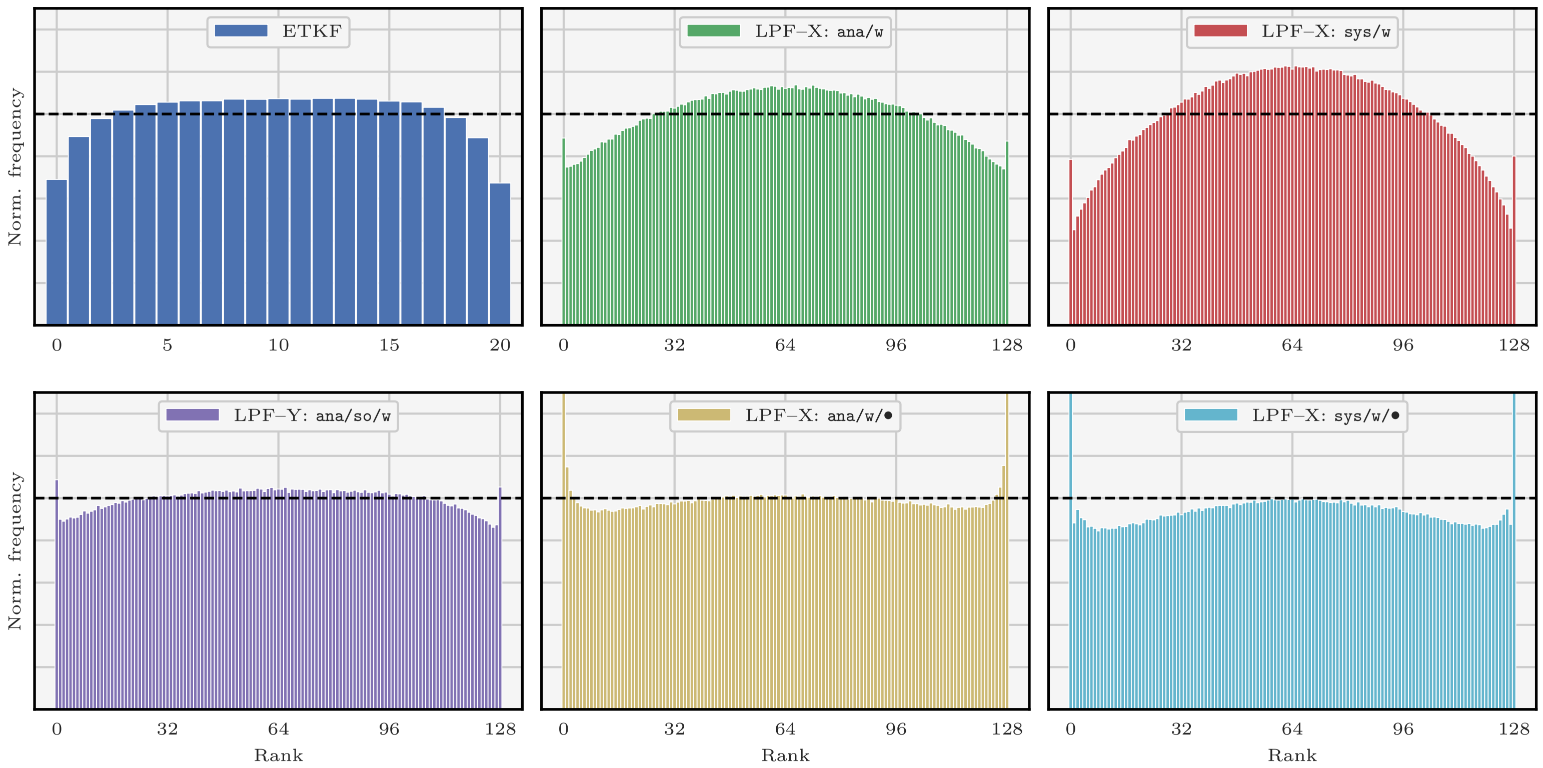

Figure 5.19: Mean rank histograms for the algorithms described in table 5.3. A dashed black line indicates the ideal frequency, $(N_\mathrm{e}+1)^{-1}$. The DA system is the L96 model in the mildly nonlinear configuration.

5.2.8 Illustration in the strongly nonlinear configuration

To conclude the L96 test series, we illustrate the performance of the LPF algorithms in the strongly nonlinear configuration of the L96 model, described in paragraph 5.1.2.3. Figure 5.20 shows the evolution of the RMSE score as a function of the ensemble size N_e for a selection of LPF–X and LPF–Y algorithms, whose implementation is described in subsections 5.2.4 and 5.2.5, as well as for the LETKF algorithm.

As expected in this strongly nonlinear configuration, the EnKF fails at accurately reconstructing the truth $\mathbf{x}^\mathrm{t}$. By contrast, all tested LPF algorithms yield, at some point, an RMSE score under the observation standard deviation $r = 1$. Regarding the ranking of the methods, most conclusions from the mildly nonlinear configuration remain true. The best RMSE scores are obtained when using optimal transport for the local updates. Combining smoothing-by-weights and colourised post-regularisation in the LPF–X algorithms yields almost equally good RMSE scores as using optimal transport for the local updates. Finally, using the LPF–Y algorithms with the second-order propagation method yields the lowest RMSE scores, despite the non-Gaussian error distributions resulting from the strong nonlinearities in this configuration.

5.3 Experiments with the BV model

In this section, we illustrate the performance of several DA algorithms using twin experiments of the BV model described in subsection 5.1.3. In subsections 5.3.1 and 5.3.2, we first consider the coarse-resolution configuration, described in paragraph 5.1.3.2. Finally in subsection 5.3.3, we consider the high-resolution configuration, described in paragraph 5.1.3.3.

For the coarse resolution configuration, in order to ensure the convergence of the statistical indicators, we use a spin-up period of $N_\mathrm{s} = 10^3$ assimilation cycles and a total simulation period of at least $N_\mathrm{c} \geq 10^4$ assimilation cycles. For the localisation in both configurations, we use the underlying physical space with the Euclidean distance. The geometry of the local blocks and domain are constructed as described in figure 4.3. Specifically, local blocks are rectangles and local domains are disks, with the difference that the doubly periodic boundary conditions are taken into account. Finally, for the LETKF algorithm, the localisation matrices $\boldsymbol{\rho}_{N_\mathrm{x}:1}$ are constructed using equation (2.57).

5.3.1 Illustration of the EnKF and of the global PF

Figure 5.21 shows the evolution of the RMSE score as a function of the ensemble size N_e for the ETKF and the LETKF algorithms. For each value of the ensemble size N_e, the multiplicative inflation factor λ, as well as the localisation radius ℓ (only for the LETKF), are optimally tuned to yield the lowest RMSE score.

The ETKF algorithm requires at least $N_\mathrm{e} = 12$ ensemble members to avoid divergence. The best RMSE scores are approximately 20 times smaller than the observation standard deviation r (0.3 in this configuration). Even with only $N_\mathrm{e} = 8$ ensemble members, the LETKF algorithm yields RMSE scores at least 10 times smaller than r, showing that, in this case, localisation is working as expected. In this configuration, the observation sites are uniformly

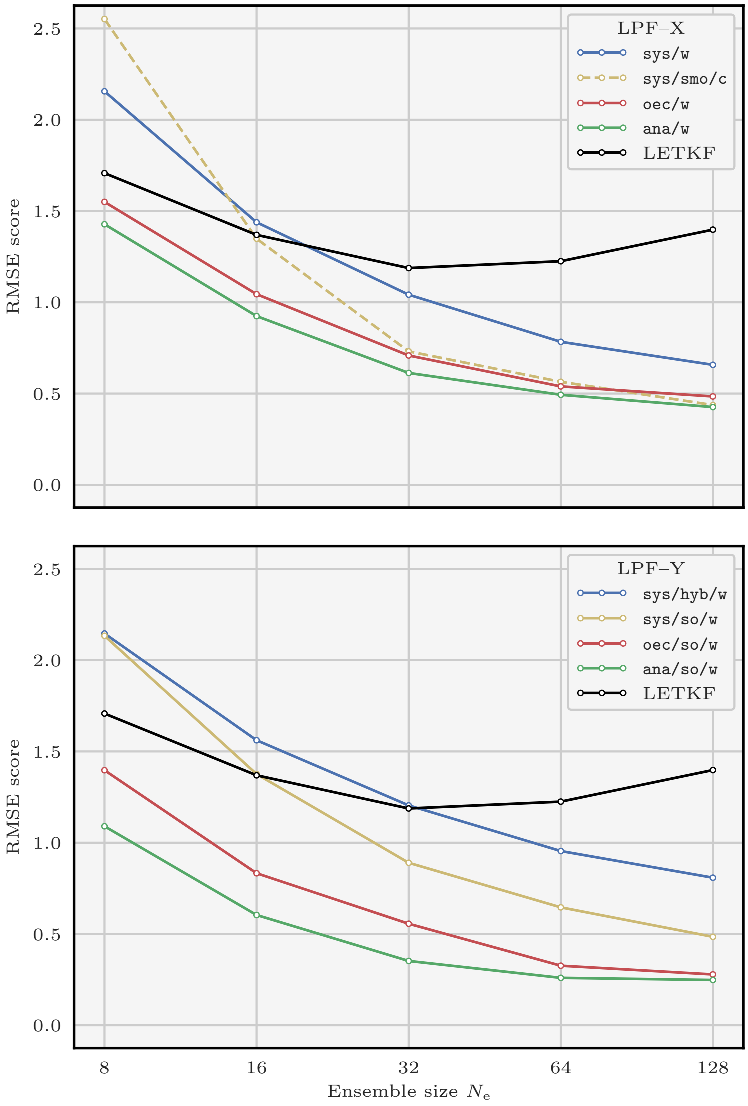

Figure 5.20: Evolution of the RMSE score as a function of the ensemble size N_e for the main LPF–X (top panel) and LPF–Y (bottom panel) algorithms. For comparison, the RMSE score of the LETKF algorithm is shown in black. The DA system is the L96 model in the strongly nonlinear configuration.

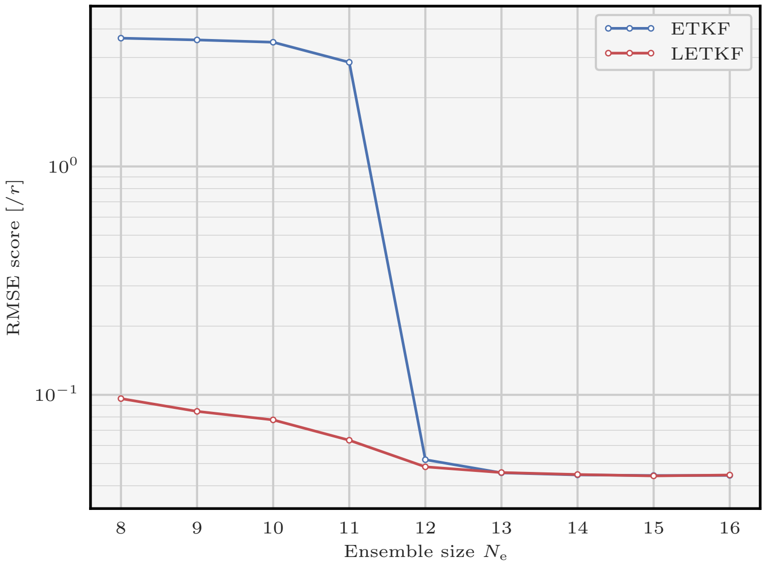

Figure 5.21: Evolution of the RMSE score as a function of the ensemble size N_e for the ETKF (in blue) and the LETKF (in red) algorithms. The DA system is the BV model in the coarse-resolution configuration, and the scores are displayed in units of the observation standard deviation r.

distributed over the spatial domain. This constrains the analysis density π^a to be close to Gaussian, which explains the success of the EnKF in this DA system.

With an ensemble of $N_\mathrm{e} \leq 1024$ particles, we could not find a combination of parameters with which the regularised SIR or ETPF algorithm yields an RMSE score significantly lower than r.

From now on, in most of the following figures related to this DA configuration, we draw a baseline at $r/20$, roughly the RMSE score of the ETKF and LETKF algorithms with $N_\mathrm{e} = 12$ ensemble members (even though slightly lower RMSE scores can be achieved with larger ensembles).

5.3.2 Illustration of the LPF algorithms

In this subsection, we test the following LPF–X and LPF–Y algorithms with an ensemble size N_e ranging from 8 to 128 particles.

- The LPF–X algorithm with systematic resampling, with or without a smoothing-by-weights step. In this case, four values for the number of local blocks N_b are tested: $N_\mathrm{b} = 1024$ local blocks with a size of 1×1 grid points, $N_\mathrm{b} = 256$ local blocks with a size of 2×2 grid points, $N_\mathrm{b} = 64$ local blocks with a size of 4×4 grid points, and $N_\mathrm{b} = 16$ local blocks with a size of 8×8 grid points, and the best RMSE score is kept. When using the smoothing-by-weights step, the smoothing strength α^s is set 1, and the smoothing radius ℓ^s is optimally tuned to yield the lowest RMSE score.

- The LPF–X algorithm with optimal ensemble coupling. In this case, we only test $N_{\mathrm{b}} = N_{\mathrm{x}} = 1024$ local blocks, and the distance radius ℓ^{d} is optimally tuned to yield the lowest RMSE score.
- The LPF–X algorithm with anamorphosis. In this case, the forecast and analysis regularisation bandwidths h^{f} and h^{a} are set to 1.
- The LPF–Y algorithm using the hybrid propagation method.
- The LPF–Y algorithm using the second-order propagation method. In this case, the local updates are performed with systematic resampling, with optimal ensemble coupling, or with anamorphosis. When using optimal ensemble coupling, the distance radius ℓ^{d} is optimally tuned to yield the lowest RMSE score, and when using anamorphosis, the forecast and analysis regularisation bandwidths h^{f} and h^{a} are set to 1.

In all cases, a post-regularisation step is added after each assimilation cycle. The localisation radius ℓ and the post-regularisation standard deviation s (for white post-regularisation) or bandwidth h (for colourised post-regularisation) are optimally tuned to yield the lowest RMSE score.

Figure 5.22 shows the evolution of the RMSE score as a function of the ensemble size N_{e} for the LPF–X and LPF–Y algorithms. Most of the conclusions for the L96 model remain true with the BV model. The best RMSE scores are obtained when using optimal transport for the local updates. Combining smoothing-by-weights and colourised post-regularisation in the LPF–X algorithms yields almost equally good RMSE scores as using optimal transport for the local updates. Finally, the lowest RMSE scores are obtained with the LPF–Y algorithms using the second-order propagation method.

With such a large model, we expected the colourised post-regularisation method to be much more effective than the white post-regularisation method, because the colourisation reduces potential spatial discontinuities in the additional jitter. However, exactly as for the L96 model, when using optimal transport for the local updates, using colourised post-regularisation does not further improve the RMSE scores. This suggests that there is room for improvement in the design of regularisation methods for PF algorithms.

Due to relatively high computational times, we restricted our study to reasonable ensemble sizes ($N_{\mathrm{e}} \leq 128$ particles). In this DA system, this is not enough for the LPF–X algorithms to yield RMSE scores comparable with those of the EnKF. However, with $N_{\mathrm{e}} \geq 64$ particles, the LPF–Y algorithms using the second-order propagation method with the anamorphosis yield RMSE scores almost equivalent to those of the EnKF.

5.3.3 Illustration in the high-resolution configuration

To conclude the BV test series, we illustrate the performance of a selection of LPF algorithms and of the LETKF algorithm in the high-resolution configuration of the BV model, described in paragraph 5.1.3.3. Using this configuration yields a higher dimensional DA system ($N_{\mathrm{x}} = 65\,536$ state variables and $N_{\mathrm{y}} = 4096$ observations) for which an assimilation cycle is too costly to perform exhaustive tests. Therefore, in this subsection, we take $N_{\mathrm{e}} = 32$ ensemble members and we monitor the time series of instantaneous RMSE score during 501 assimilation steps, which corresponds to a total simulation time of 250 time units.

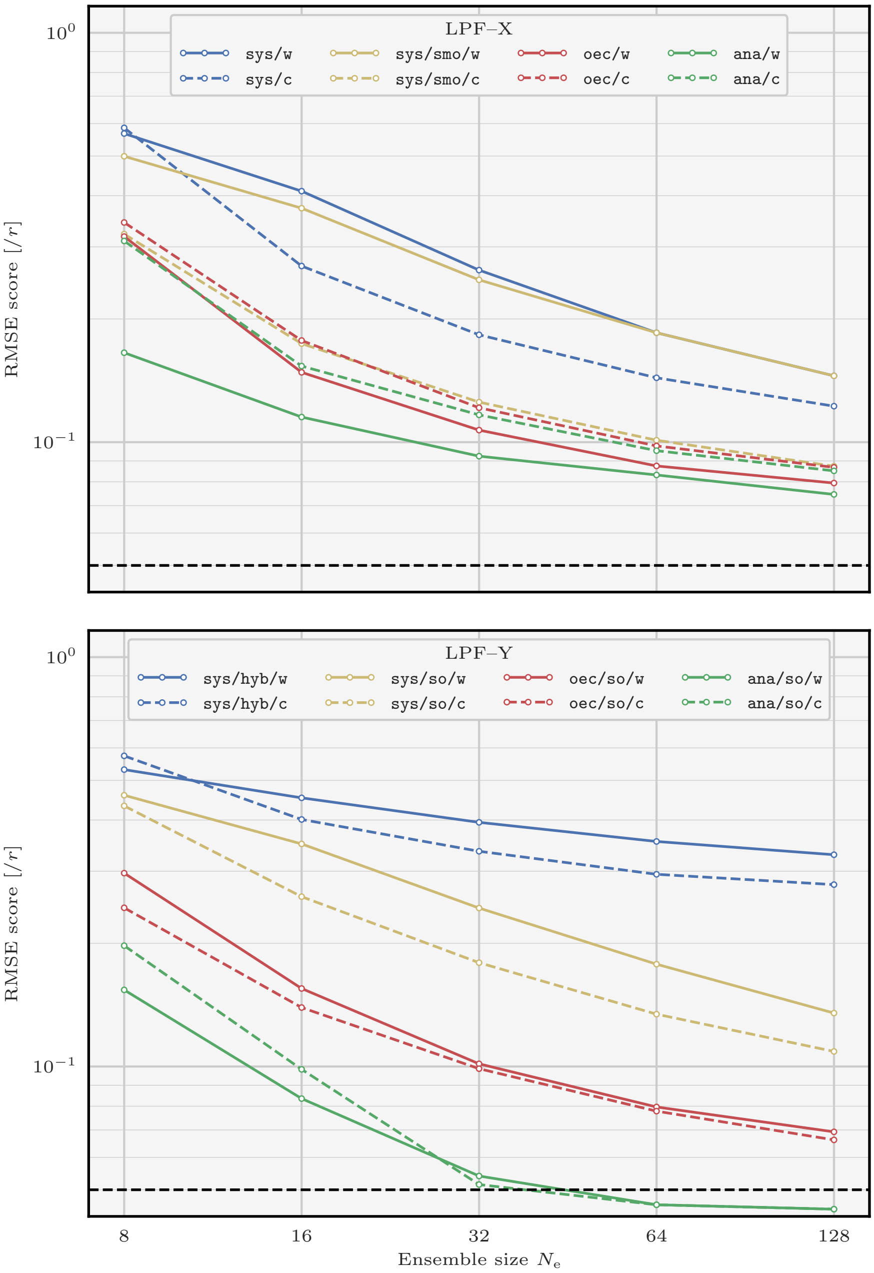

Figure 5.22: Evolution of the RMSE score as a function of the ensemble size N_e for the LPF–X (top panel) and the LPF–Y (bottom panel) algorithms. For comparison, the RMSE score of the ETKF algorithm with $N_\mathrm{e} = 12$ members is shown with an horizontal dashed black line. The DA system is the BV model in the coarse-resolution configuration, and the scores are displayed in units of the observation standard deviation r.

Table 5.4: Characteristics of the DA algorithms tested with the BV model in the high resolution configuration. The first column indicates the corresponding label in figure 5.23 following the conventions of tables 5.1 and 5.2. The localisation radius ℓ is given in units of the simulation domain $L = 1$. Furthermore, all LPF–X algorithms use $N_{\mathrm{b}} = N_{\mathrm{x}}$ local blocks. The reported RMSE score in the fifth column is averaged over the final 300 assimilation cycles and is given in units of the observation standard deviation r. The wall-clock simulation time reported in the sixth column is the average time spent per analysis step. For comparison, the average time spent per forecast (for a time interval between consecutive observation Δt of 0.5 unit of time) for the 32-member ensemble is 0.94 s. When possible, parallelisation is enabled in the N_{x} local updates using 24 `OpenMP` threads. The average time spent per analysis step for the parallelised runs, as well as the acceleration factor, are reported in the last column.

Label	ℓ/L	s, h	Other param.	RMSE/r	Wall-clock time	
					1 thread	24 threads
LETKF	0.35	–	$\lambda = 1.04$	0.10	103.90	5.09 (×20.41)
LPF–X: `sys/w`	0.02	0.55	–	0.78	7.58	0.54 (×14.04)
LPF–X: `sys/smo/c`	0.05	1.00	$\alpha^{\mathsf{s}} = 1$, $\ell^{\mathsf{s}} = \ell$	0.38	226.20	12.50 (×18.10)
LPF–X: `ana/w`	0.08	0.11	$h^{\mathsf{f}} = h^{\mathsf{a}} = 3$	0.33	13.94	0.86 (×16.21)
LPF–Y: `sys/hyb/w`	0.03	0.70	–	0.90	122.18	–
LPF–Y: `sys/so/w`	0.07	0.25	–	0.46	52.97	–
LPF–Y: `ana/so/w`	0.20	0.01	$h^{\mathsf{f}} = h^{\mathsf{a}} = 1$	0.13	64.79	–

For these experiments, the selection of algorithms is listed in table 5.4. Each algorithm uses the same initial ensemble $\mathbf{E}$ obtained as follows:

$$\forall i \in (N_{\mathrm{e}} : 1), \quad \mathbf{x}_i(0) = \mathbf{x}^{\mathsf{t}}(0) + 0.5 \times \mathbf{e} + \mathbf{e}_i, \quad (\mathbf{e}, \mathbf{e}_i) \sim \mathcal{N}[\mathbf{0}, \mathbf{I}]. \tag{5.20}$$

Such an ensemble is not very close to the truth $\mathbf{x}^{\mathsf{t}}$ (as measured by the RMSE), and its spread is large enough to reflect the lack of initial information. Approximate values for the localisation radius ℓ, for the post-regularisation standard deviation s (when using white post-regularisation), for the post-regularisation bandwidth h (when using colourised post-regularisation), and for the multiplicative inflation (when using the LETKF algorithm) are found using several twin experiments with a few hundred assimilation cycles (not illustrated here). When using anamorphosis, we only test a few values for the forecast and analysis regularisation bandwidths h^{f} and h^{a}, when using the smoothing-by-weights step, the smoothing strength α^{s} is set to 1 and the smoothing radius ℓ^{s} is set to be equal to the localisation radius ℓ. Finally, all LPF–X algorithms are tested with $N_{\mathrm{b}} = N_{\mathrm{x}} = 65\,536$ local blocks.

Figure 5.23 shows the evolution of the instantaneous RMSE score for the selected algorithms. All experiments are performed on the same computational platform with 12 cores. The algorithmic parameters, alongside the average RMSE score, computed over the final 300 assimilation steps and the wall-clock computational times, are reported in table 5.4. In terms of RMSE score, the ranking of the algorithms is unchanged, and most of the conclusions for this experiment are the same as with the coarse resolution configuration.

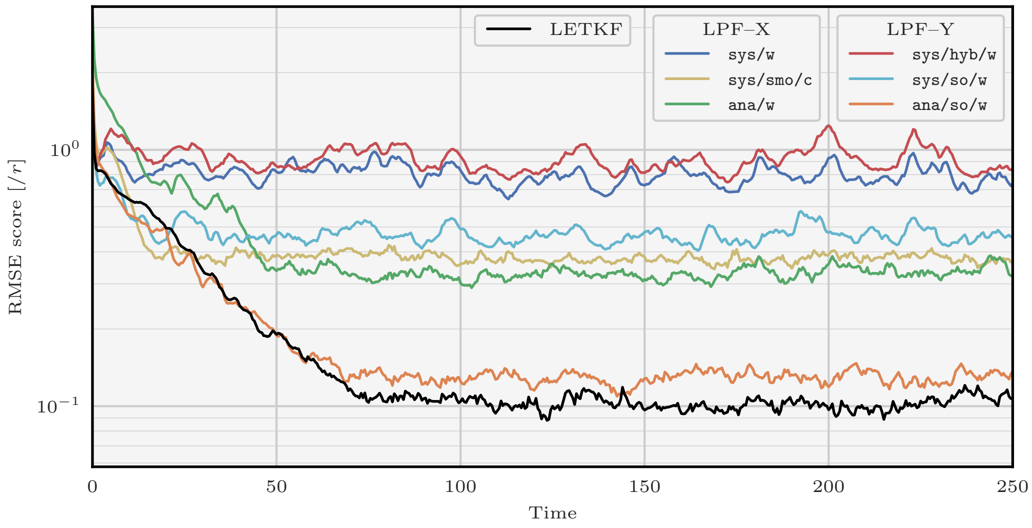

Figure 5.23: Time series of instantaneous RMSE score for the algorithms described in table 5.4. The DA system is the BV model in the high-resolution configuration, and the scores are displayed in units of the observation standard deviation r.

Thanks to the uniformly distributed observation network, the analysis density π^{a} is close to Gaussian. Therefore the LETKF algorithm can efficiently reconstruct a good approximation of the truth $\mathbf{x}^{\mathsf{t}}$. As expected in this high-dimensional DA configuration, the algorithms using a second-order truncation (the LETKF algorithm and the LPF–Y algorithms using the second-order propagation method) are more robust. Optimal values of the localisation radius ℓ are qualitatively large, which allows for a better reconstruction of the system dynamics.

For the LPF–X algorithm with systematic resampling as well as for the LPF–Y using the hybrid propagation method, ℓ needs to be very small to counteract the curse of dimensionality. With such small values for ℓ, the local domains contains only 4 to 13 observation sites, which is empirically barely enough to reconstruct $\mathbf{x}^{\mathsf{t}}$ with an RMSE score lower than the observation standard deviation r. As in the previous experiments, using optimal transport for the local updates or combining smoothing-by-weights and colourised post-regularisation yields significantly lower RMSE scores. The RMSE scores of the LPF–X algorithm with anamorphosis and of the LPF–X algorithm with systematic resampling, though not as good as the RMSE score of the LETKF algorithm, show that $\mathbf{x}^{\mathsf{t}}$ is reconstructed with an acceptable accuracy. The RMSE scores of the LETKF algorithm and of the LPF–Y algorithm using the second-order propagation method with anamorphosis are almost comparable. Depending on the algorithm, the conditioning to the initial ensemble $\mathbf{E}$ more or less quickly vanishes.

Without parallelisation, we observe that the N_{x} local updates of the LPF–X algorithms are almost always faster than both the N_{x} local analyses of the LETKF algorithms and the N_{y} sequential updates of the LPF–Y algorithms. The second-order propagation method is slower because of the linear algebra involved in the method and the hybrid propagation algorithm is slower because computing the prior and posterior update weights ω^{f} and ω^{a} is numerically demanding. The LETKF algorithms is slower because of the matrix inversions in ensemble space. Finally, the LPF–X algorithm with smoothing-by-weights is even slower because computing the smoothed ensemble in this two-dimensional model is numerically demanding. The difference between the LPF–X and LPF–Y algorithms is even more visible in the parallelised runs. The LPF–Y algorithms are not parallel, which is why they are more than 70 times slower than the fastest LPF–X algorithms.

5.4 Summary and discussion

In this chapter, we have implemented and systematically tested the LPF algorithms using twin experiments of the L96 and the BV models. With these models, implementing localisation is simple and works as expected: the LPF algorithms yield acceptable RMSE scores, even with small ensembles, in regimes where global PF algorithms are degenerate. In terms of RMSE scores, there is no clear advantage of using the hybrid propagation method (designed to avoid unphysical discontinuities) over the simpler LPF–X algorithms, which have a lower computational cost. As expected, algorithms based on the second-order propagation method are less sensitive to the curse of dimensionality and yield the lowest RMSE scores. We have shown that using optimal transport for the local updates always yields important gains in RMSE score. For the LPF–X algorithms, this is a consequence of mitigating the unphysical discontinuities introduced while assembling the locally updated particles. For the LPF–Y algorithms, this is a consequence of the minimisation of the local updates to be propagated.

The successful application of the LPF algorithms to DA systems with a perfect model is largely due to the use of post-regularisation. Using post-regularisation introduces an additional bias in the analysis alongside an extra parameter to be determined. For our numerical experiments, we have introduced two post-regularisation methods: white post-regularisation, in which the Gaussian jitter is a white noise, and colourised post-regularisation, in which the Gaussian jitter has covariance matrix determined by the ensemble. We have discussed the relative performance of each method and concluded that there is room for improvement in the design of regularisation methods for PF algorithms. Ideally, the regularisation methods should be adaptive and built concurrently with the localisation method.

The local update method is the main ingredient in the success, or failure, of an LPF algorithm. The approaches based on optimal transport offer an elegant and efficient framework to deal with the discontinuity issue inherent to the local updates. However, the algorithms derived in this article could be improved. For example, it would be desirable to avoid the systematic reduction to one-dimensional problems when using the anamorphosis.

The successful application of the LPF algorithms to the BV model in the high-resolution configuration shows that the algorithms may be ready to be applied to realistic DA systems. This is the topic of chapter 6. Finally, the localisation frameworks introduced in chapter 4 can only work with local observations. The ability to assimilate non-local observations becomes increasingly important with the prominence of satellite observations. This topic is discussed in chapter 7 in an EnKF context.

6 Application to the prediction of ozone at continental scale

Contents

In chapter 5, the performances of the LPF algorithms have been illustrated using twin experiments with low- and medium-order DA systems. In this chapter, we consider the case study of the prediction of the tropospheric ozone (O_3) concentration in western Europe during the summer 2009. Ozone is one of the most regulated pollutants in Europe (and all over the world as well) because it damages human health. Although it is not directly emitted, it is found in high concentrations in urban areas, as a secondary product of photochemistry.

The evolution of the ozone concentrations in the atmosphere is determined by the equations of atmospheric chemistry. Such equations are usually solved using a chemistry and transport model (CTM), in which the meteorology is not directly computed, but used as input of the model. The experiments described in this chapter are a continuation of the work of Haussaire (2017), in which we use the CTM `Polair3DChemistry` from the `Polyphemus` framework (Mallet et al. 2007). By contrast with the work presented in the other chapters, this work is not yet *complete*, meaning that several aspects of the study still need to be explored.

Section 6.1 presents the numerical experiments performed in this chapter. In particular, we describe the observation database and the dynamical model. Section 6.2 describes the implementation of the DA algorithms. The experiments and their results are then discussed in section 6.3. Finally, conclusions are provided in section 6.4.

6.1 Presentation of the numerical experiments

6.1.1 The observation database

`Airbase` is an air quality database managed by the European Environment Agency. It gathers the observations of several pollutants, among which ozone but also carbon dioxide (CO_2) and nitrogen dioxide (NO_2), at several stations spread all over Europe. Most of the stations are located in the physical domain $\mathcal{D}$ defined by:

$$\mathcal{D} = [9°\mathrm{W}, 24°\mathrm{E}] \times [36°\mathrm{N}, 59°\mathrm{N}]. \tag{6.1}$$

Each station has a *type*, which informs about the nature of the pollutant source. It can be *industrial*, *traffic*, or *background*. Each station also have an *ozone class*, which informs about the population density. It can be *urban*, *suburban*, or *rural*. From all these stations, we only keep the background rural stations, because the resolution of the simulation considered in this chapter is not enough to capture phenomenon at a more local scale.

In this chapter, we only consider the measurements of ozone concentration. Figure 6.1 shows the repartition of the stations in the database, which are used in the DA experiments of section 6.3. At each station, there are 24 measurements per day. They correspond to the average ozone concentration over time periods of 1 h, starting at 0:00 UTC. For the DA experiments, we make the approximation that each measurement corresponds to the instantaneous ozone concentration at the centre of the time period (that is, the first measurement of each day is assumed to be the instantaneous ozone concentration at 0:30 UTC). In summary, the time interval between consecutive observations Δt is 1 h, and the number of observations N_y is equal to the number of stations.

Remark 23. The measurements are not always available. In the DA experiments, when the measurement at a given time and at a given station is not available, the station is simply disabled for the assimilation cycle.

6.1.2 Description of the model

The atmosphere is composed of dinitrogen (N_2, about 80 %), of dioxygen (O_2, about 20 %), and of many other chemical species, among which ozone. The evolution of the concentration field c_s of each chemical species s in the atmosphere is governed by the following advection-diffusion-reaction equation:

$$\frac{\partial c_s}{\partial t} + \mathrm{div}(\mathbf{v}c_s) = \mathrm{div}(\mathbf{K}\nabla c_s) + \chi_s(\mathbf{c}, T, J, h) + S_s - \Lambda_s c_s. \tag{6.2}$$

The goal of a CTM is to solve this equation of all chemical species using input data coming from several sources:

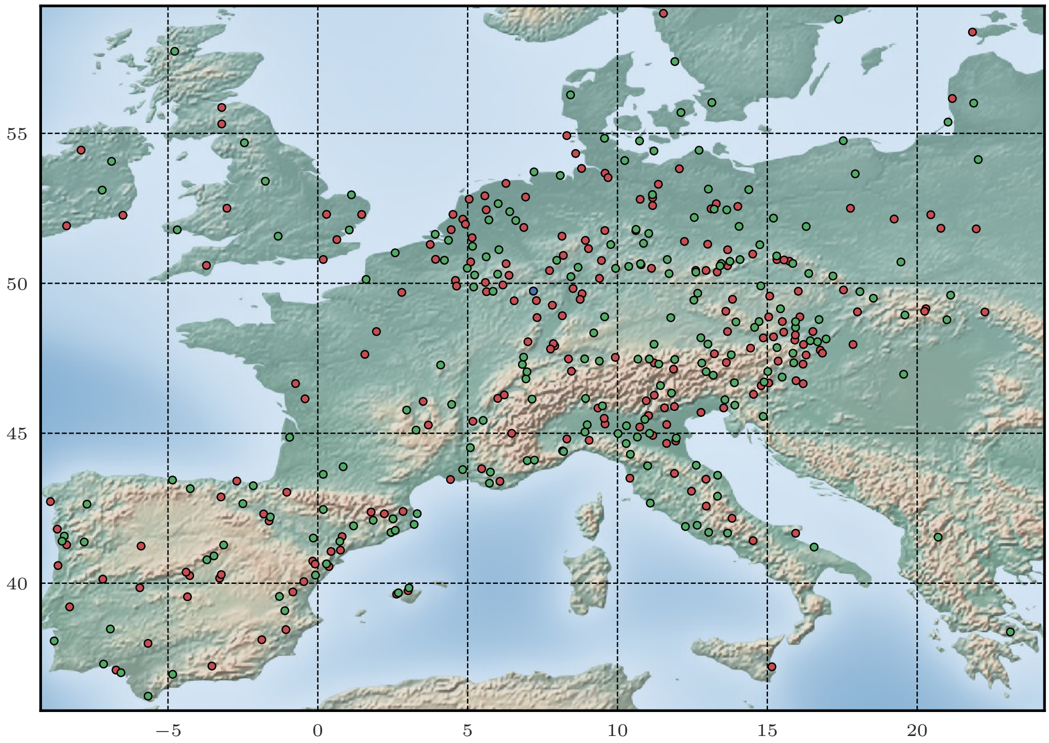

Figure 6.1: Simulation domain $\mathcal{D}$. The red, blue, and green dots show the N_{y} observation sites, *i.e.*, the background rural stations which are used in the DA experiments of section 6.3. According to the nomenclature introduced in subsection 6.2.1, the red stations form the group `assim` and the green and blue stations form the group `valid`. The blue station is the 10^{th} station of the group `valid`.

Table 6.1: Vertical limits for all $P_z = 11$ vertical levels.

Level	Lower boundary [m]	Upper boundary [m]
1	0	40
2	40	90
3	90	180
4	180	320
5	320	600
6	600	1000
7	1000	1400
8	1400	1900
9	1900	2400
10	2400	3000
11	3000	5000

- the meteorological data, that is the fluid velocity vector field $\mathbf{v}$, the effective diffusion matrix $\mathbf{K}$, the temperature field T, the photolysis rates field J, and the specific humidity field h;
- the emission data, that is the source term for each species S;
- the physical parametrisation of the chemistry, that is the production rate of each species χ, which in particular depends on the concentration of all species $\mathbf{c}$, and the deposition rate of each species Λ.

Computing the production rates χ is the core element of a CTM, and involves the description of several hundreds of chemical reactions between all species. Different chemical mechanisms exist, depending on the choice of the species and reactions described.

In the atmosphere, ozone is mainly produced by the photolysis of nitrogen dioxide. Nitrogen dioxide itself is produced by oxidation of nitrogen oxide by radicals, which are in turn produced by oxidation and photolysis of volatile organic compounds. Hence, the ozone concentration is highly related to the concentration of nitrogen oxide and dioxide, and to the concentration of volatile organic compounds. Therefore, we need to select a chemical mechanism which describes at least all these elements.

For our experiments, we use the `Polair3DChemistry` model (Mallet et al. 2007), with a parametrisation briefly described in the following lines. The simulation domain is a discretisation of the domain $\mathcal{D}$ using a resolution of $0.5° \times 0.5°$ in latitude and longitude coordinates. There are $P_z = 11$ vertical levels whose boundaries are reported in table 6.1. The chemistry is described using the `CB05` mechanism without aerosols (Yarwood et al. 2005). It describes the evolution of $P_s = 52$ gaseous species, among which ozone.

The numerical integration of equation (6.2) is performed using a first-order splitting, with an integration time step δt of 600 s. The advection is solved using a third-order direct space time scheme, and a Koren-Sweby flux limiter. The diffusion and the chemistry are solved using a second-order Rosenbrock method, with an adaptive time step in the range $[10\,\mathrm{s}, 600\,\mathrm{s}]$.

Table 6.2: Summary of the parametrisations used in the CTM `Polair3DChemistry`.

Period	Summer 2009
Resolution	0.5° × 0.5°
Domain (latitude)	9°W – 24°E ($P_x = 67$)
Domain (longitude)	36°N – 59°N ($P_y = 47$)
Vertical levels	table 6.1 ($P_z = 11$)
Chemical mechanism	`CB05` ($P_s = 52$)
Aerosols	disabled
Meteorology	ECMWF, 3 h forecast
Initial conditions	MOZART `2.0`
Boundary conditions	MOZART `2.0`
Anthropogenic emissions	EMEP
Biogenic emissions	MEGAN
Vertical diffusion	Louis (1979)
Horizontal diffusion	10^5 m^2/s
Deposition	Zhang et al. (2003)

The meteorological input fields are given by the three-hour forecast fields of the European Centre for Medium-Range Weather Forecasts (ECMWF). The initial and boundary conditions are extracted from global simulations using the second version of the Model for OZone And Related chemical Tracers (MOZART, Horowitz et al. 2003). The anthropogenic emissions come from the European Monitoring and Evaluation Programme (EMEP, Vestreng et al. 2004), and the biogenic emissions from the Model of Emissions of Gases and Aerosols from Nature (MEGAN) (MEGAN, Guenther et al. 2006), using the land use data from the Global Land Cover Facility (GLCF). The horizontal diffusion coefficient is set to 10^5 m^2/s, and the vertical diffusion coefficients are computed using the parametrisation of Louis (1979). Finally, we use the deposition model of Zhang et al. (2003). The full list of parametrisations is summarised in table 6.2.

6.1.3 The reference simulation

A reference simulation is performed using the parametrisations described in the previous subsection, from the 1st May 2009 to the 31st August 2009. We take a spin-up period of one entire month to relax the influence of the initial conditions.

The instantaneous ozone concentration predicted by the simulation at the q-th station is computed as

$$y^{\mathrm{s}}_q(t) = \mathbf{H}_q \mathbf{x}^{\mathrm{s}}(t), \tag{6.3}$$

where $\mathbf{x}^{\mathrm{s}}(t)$ is the vector with $P_z \times P_y \times P_x$ elements representing the instantaneous ozone field predicted by the simulation in the full domain, and $\mathbf{H}_q$ is the bilinear interpolation matrix for the q-th station. This instantaneous predicted ozone concentration $\mathbf{y}^{\mathrm{s}}(t)$ is compared to the instantaneous measured ozone concentration $\mathbf{y}(t)$ during the whole summer period (June, July, and August). To limit as much as possible the impact of a possible time delay, the instantaneous predicted concentration is taken at the center of the corresponding average measurement period (for example the average measured concentration from 0:00 to 1:00 UTC is compared to the instantaneous predicted concentration at 0:30 UTC). We have checked that taking the average predicted concentration, with prediction records every ten minutes, does not change our conclusions.

This comparison shows that the ozone concentration predicted by the reference simulation is biased and yields high RMSE scores (see the detailed comparison in paragraph 6.1.4.3). This can be explained by two factors:

- the selected model is crude, and in particular it does not incorporate the aerosols;
- the input database may be biased.

In order to overcome this limitation, the reference simulation is debiased, as presented in the following subsection.

For completeness, it should be mentioned that the choice of a bilinear interpolation method is convenient, because the resulting observation operator $\mathbf{H}$ is linear and sparse. Furthermore, we have checked that using a higher order interpolation method barely changes the time series of predicted ozone concentration. Therefore, we only consider the bilinear interpolation method in this chapter.

6.1.4 Debiasing the reference simulation

The idea of debiasing the simulation is to modify the observation operator in such a way that the instantaneous ozone concentration predicted at the q-th station is now computed as

$$y_q^{\mathsf{s}}(t) = \mathbf{H}_q \mathbf{x}^{\mathsf{s}}(t) - b_q(t), \tag{6.4}$$

where $b_q(t)$ is an instantaneous bias parameter for the q-th station. It must be chosen to match the instantaneous bias $\mathbf{H}_q \mathbf{x}^{\mathsf{s}}(t) - y_q(t)$ while being as simple as possible.

6.1.4.1 A simple parametrisation for the bias

We choose to represent the bias parameter for the q-th station as

$$\forall (d,h) \in \mathbb{N} \times (24:1), \quad b_q\big(t(d,h)\big) \triangleq \lambda_q \gamma_h + \mu_q, \tag{6.5}$$

where d is the index of the simulation day, h is the hour of the day, and $t(d,h)$ is the corresponding time. In other words, the bias parameter for the q-th station is derived from a common daily cycle $\gamma_{24:1}$ using a stretching parameter λ_q and a level parameter μ_q, which are constant in time. This means that we have to determine the value of 2 bias parameters per station and of 24 common bias parameters.

For simplicity, we define the vectors $\boldsymbol{\gamma}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ as the vector whose components are $\gamma_{24::}$, $\lambda_{N_{\mathrm{y}}:1}$, and $\mu_{N_{\mathrm{y}}:1}$, respectively.

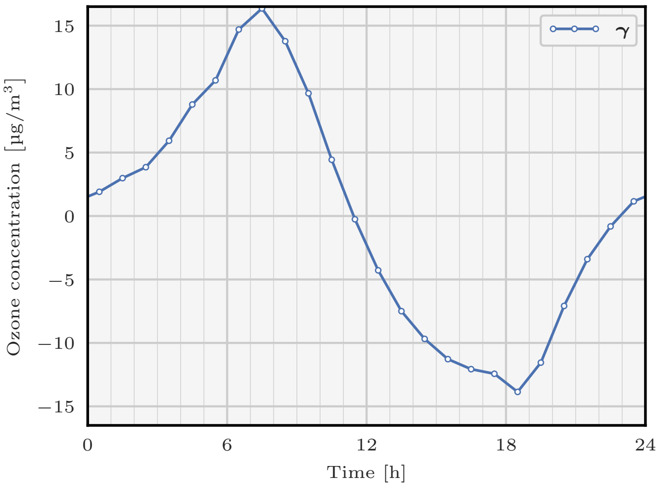

Figure 6.2: Common daily cycle of bias $\boldsymbol{\gamma}$.

6.1.4.2 Calibration of the bias parameters

The values for the bias parameters are computed by minimising the (masked) cost function

$$\mathcal{J}(\boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \sum_{d \in \mathcal{D}_\mathrm{c}} \sum_{h=1}^{24} \sum_{q=1}^{N_\mathrm{y}} \mathbf{1}^{\mathrm{o}}_{d,h,q} \big[\mathbf{H}_q \mathbf{x}^\mathrm{s}\big(t(d,h)\big) - y_q\big(t(d,h)\big) - \lambda_q \gamma_h - \mu_q \big]^2, \tag{6.6}$$

where $\mathcal{D}_\mathrm{c}$ is the set of days selected for the calibration, and $\mathbf{1}^{\mathrm{o}}_{d,h,q}$ is a factor equal to 1 if the measurement at the q-th station, the d-th day and the h-th hour is available, and equal to 0 otherwise.

In our case, there is a total of 92 days for the summer period, which we divide into batches of 10 days. The first batch, as well as every other batch, are placed into $\mathcal{D}_\mathrm{c}$. All remaining batches, as well as the last remaining two days, are placed into $\mathcal{D}_\mathrm{v}$, the set of days selected for the validation of the bias parameters. Hence, $\mathcal{D}_\mathrm{c}$ contains 50 days and $\mathcal{D}_\mathrm{v}$ contains 42 days.

The cost function $\mathcal{J}$ is a nonlinear function of $2N_\mathrm{y} + 24$ variables. However its gradient is easy to compute and standard minimising algorithms such as the L-BFGS-B algorithm (Byrd et al. 1995) only require a few dozen iterations to find its minimum.

For any $\alpha \in \mathbb{R}$, using the set of bias parameters $(\boldsymbol{\gamma} + \alpha \mathbf{1}, \boldsymbol{\lambda}, \boldsymbol{\mu} - \alpha \boldsymbol{\lambda})$ is equivalent to using the set of bias parameters $(\boldsymbol{\gamma}, \boldsymbol{\lambda}, \boldsymbol{\mu})$. Therefore, we force the daily cycle $\boldsymbol{\gamma}$ to have a zero average value. The result of the minimisation is depicted in figures 6.2, and 6.3. For completeness, we have minimised the cost function $\mathcal{J}$ in the cases where the calibration period $\mathcal{D}_\mathrm{c}$ contains every other day in the summer period, or simply every day in the summer period. In both cases, the result of the values for the bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ (not illustrated here) are very similar to those depicted in figures 6.2, and 6.3. This shows that the calibration of the bias parameters is adequate.

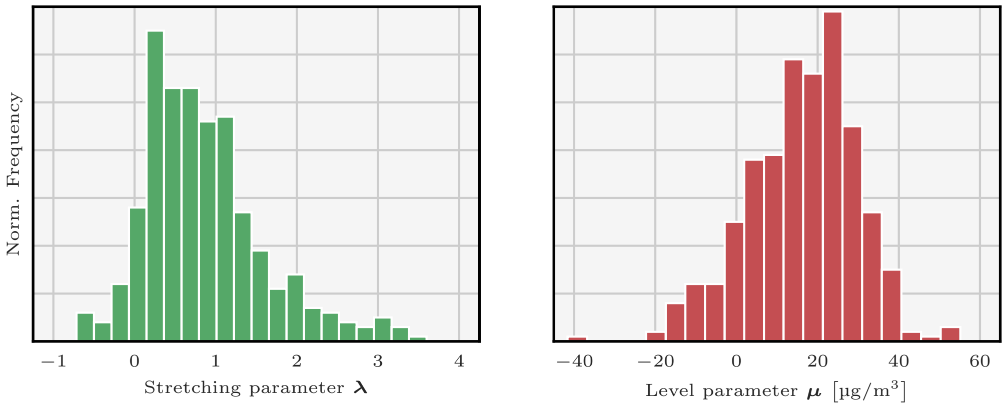

Figure 6.3: Distribution of the stretching parameter $\boldsymbol{\lambda}$ (left panel) and of the level parameter $\boldsymbol{\mu}$ (right panel) over the list of stations.

Table 6.3: Average statistical indicators per station for the original (middle column) and debiased (right column) reference simulations.

Indicator		Original simulation	Debiased simulation
Mean bias	[µg/m^3]	16.40	0.52
RMSE	[µg/m^3]	29.84	20.08
Correlation		0.54	0.63

6.1.4.3 Validation of the bias

Figure 6.4 shows the time series of average ozone concentration, where at each time the average is taken over all stations for which a measurement is available, and figure 6.5 shows the average daily cycle of average ozone concentration. In both cases, the debiased predictions are much closer to the measurements than the original ones.

The effect of the debiasing in the reference simulation is then quantified by computing the following statistical indicators. For each station, we compute the mean bias, the RMSE, and the correlation between the time series of predicted and measured ozone concentration during the validation period $\mathcal{D}_\mathrm{v}$ only. The average values are reported in table 6.3, where in each case the average is taken over all stations. Thanks to the debiasing, the mean bias has been almost entirely removed, and the other statistical indicators (RMSE and correlation) now have the same order as for typical CTMs (Bessagnet et al. 2016).

6.2 Implementation of data assimilation

For the numerical experiments of this chapter, five different DA algorithms are used:

- the cycled BLUE algorithm, algorithm 1.1, described in subsection 1.5.2, and which is

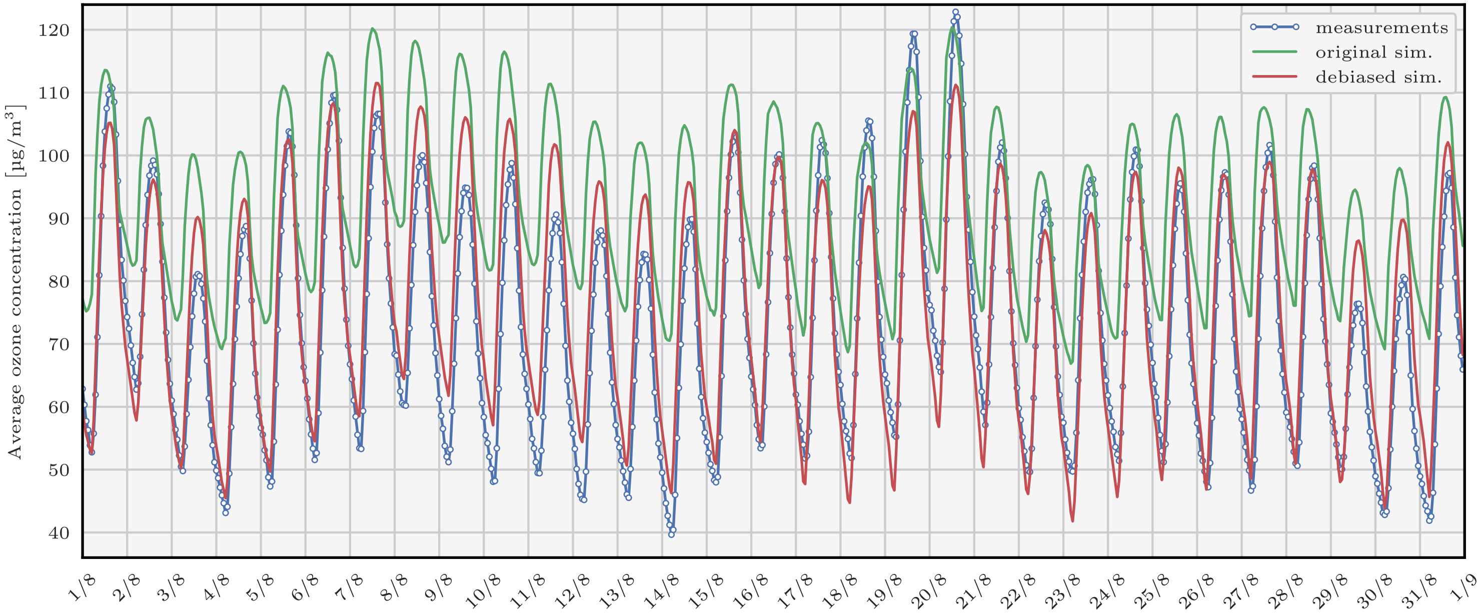

Figure 6.4: Time series of average ozone concentration during the month of August for the measurements (in blue), for the original reference simulation (in green), and for the debiased reference simulation (in red).

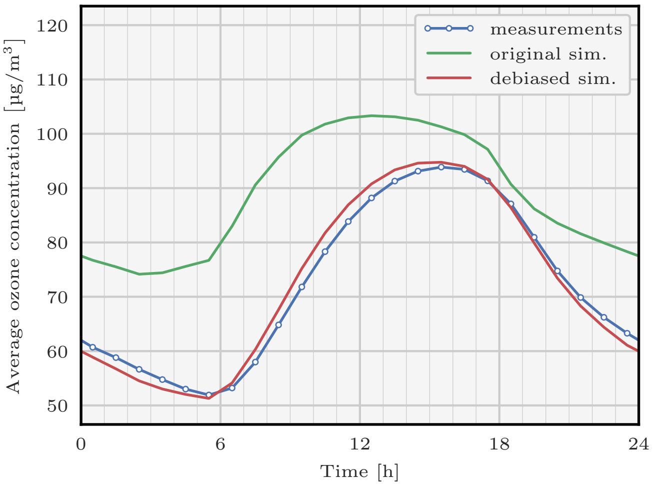

Figure 6.5: Average daily cycle of average ozone concentration for the measurements (in blue), for the original reference simulation (in green), and for the debiased reference simulation (in red).

also called optimal interpolation (OI) algorithm in this chapter;

- the LETKF algorithm, algorithm 2.4, described in subsection 2.5.4;
- the LPF–X algorithm, algorithm 4.1, in which the local updates are performed using the adjustment-minimising systematic resampling algorithm (algorithm 3.4 with the modification described in paragraph 4.2.3.3);
- the LPF–X algorithm, algorithm 4.4, in which the local updates are performed using the anamorphosis, as described in paragraph 4.2.3.5;
- the LPF–Y algorithm, algorithm 4.8, in which the local updates are performed using the adjustment-minimising systematic resampling algorithm and propagated using the second-order propagation method, as described in subsection 4.3.3.

In this section, we describe how these algorithms are implemented for the assimilation of the observations in the database described in subsection 6.1.1.

6.2.1 General considerations

In the DA experiments, we chose to keep in the control vector $\mathbf{x}$ all chemical species at all vertical levels. In other words, there are $N_\mathrm{x} = P_\mathrm{s} \times P_\mathrm{z} \times P_\mathrm{y} \times P_\mathrm{x} = 1\,801\,228$ variables in this configuration. This choice enables the development of interspecies correlations during the DA experiment, which means that the analysis step will be able to correct the concentrations of all chemical species, even though only ozone is observed. Other choice are possible (see, *e.g.*, Gaubert 2013, and references therein).

In the observation database, the observations are available every $\Delta t = 1\,\mathrm{h}$. The observation operator $\mathbf{H}$ is the bilinear interpolation operator introduced in subsection 6.1.3, with the debiasing method described in subsection 6.1.4. The bias parameters $\boldsymbol{\gamma}$, $\boldsymbol{\lambda}$, and $\boldsymbol{\mu}$ could have been included in the control vector to be estimated by the DA algorithms. However, preliminary experiments have shown that this barely improves the scores,[1] which is why we have chosen to keep the values of the bias parameters static. For simplicity, the observation error covariance matrix $\mathbf{R}$ is chosen to be diagonal:

$$\mathbf{R} \triangleq r^2\mathbf{I}, \tag{6.7}$$

where the observation standard deviation r need to be specified. Typical values of the observation standard deviation r lie in the interval $\left[10\,\mu\mathrm{g/m^3}, 20\,\mu\mathrm{g/m^3}\right]$ (see, *e.g.*, Haussaire 2017, and references therein).

The 1 h forecast between consecutive observations are performed using the `Polair3DChemistry` model with the parametrisation described in subsection 6.1.2. The resulting dynamical model $\mathcal{M}$ is expected to be stable (non-chaotic), but highly nonlinear as a result of the chemical processes in the atmosphere. For the ensemble DA algorithms (the LETKF and the LPF algorithms here), the forecast of each individual ensemble member is performed independently on a separate `OpenMP` thread of the computational platform. As a consequence, the wall-clock time for the ensemble forecast is equivalent to the (non-parallelised) wall-clock time for a single member forecast. For the OI algorithm, we only need to forecast one state. Therefore in this case, parallelisation is enabled in the `Polair3DChemistry` model under the form of `OpenMP` threads.

The DA experiments consist of 5 days of assimilation, starting from the 14th August 2009 at 0:30 UTC. This corresponds to a total of 119 forecast steps, and 120 assimilation steps. Finally, note that these 5 days of assimilation are not included in the 50 days used to calibrate the bias parameters. This means that the statistical information contained in the observations is not used twice. From all stations available in `Airbase`, we only keep we only keep those for which less than 25 % of measurements are missing during the DA period. The stations are then randomly divided into two groups. The first group (`assim`) contains the observations which are assimilated during the experiments. The second group (`valid`) contains the observation which are used for cross-validation. Both groups contains $N_{\mathrm{y}} = 202$ stations, and are shown in figure 6.1.

Finally, it should be mentioned that concentration values are expected to be non-negative. The analysis step of the DA algorithms may not respect this condition, which is why, after each assimilation cycle, negative values for the concentration are cropped to zero.

6.2.2 The optimal interpolation algorithm

In addition to the general considerations, we only need to specify the background error covariance matrix $\mathbf{B}$ to implement the OI algorithm. Following the choice of Haussaire (2017), the covariances are non-null only for the ozone concentrations, and the background error

[1] To be more specific, we have observed that, when the bias parameters are included in the control vector, the DA algorithms systematically tend to correct the bias parameters at the expense of the concentration variables. This is most probably due to a bad specification of the bias uncertainty.

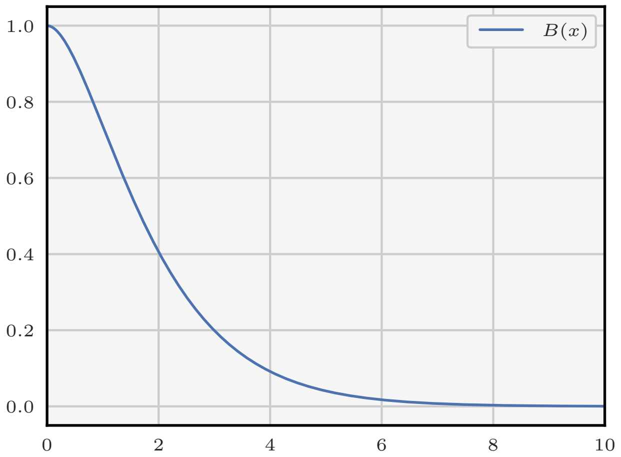

Figure 6.6: Balgovind function, defined by equation (6.9), in blue.

covariance matrix $\mathbf{B}_{\mathrm{O}_3}$ for the ozone has (m_z, m_y, m_x)-th row, (n_z, n_y, n_x)-th column element given by

$$[\mathbf{B}_{\mathrm{O}3}]_{(m_z,m_y,m_x),(n_z,n_y,n_x)} = p^2\, \delta_{m_z \le z_{\mathrm{bl}}}\, \delta_{n_z \le z_{\mathrm{bl}}}\, B\left(\frac{d_{m_z,n_z}}{\ell_{\mathrm{v}}}\right) B\left(\frac{d_{(m_y,m_x),(n_y,n_x)}}{\ell_{\mathrm{h}}}\right), \qquad (6.8)$$

where B is the Balgovind correlation function (Balgovind et al. 1983), defined by

$$B : \begin{cases} \mathbb{R}_+ & \to \mathbb{R}_+, \\ x & \mapsto x\exp(-x), \end{cases} \qquad (6.9)$$

and illustrated in figure 6.6. In equation (6.8), the following quantities are used:

- d_{m_z,n_z} is the (vertical) distance between the m_z-th and the n_z-th levels, and ℓ_{v} is the vertical correlation radius, set to 3000 m;
- $d_{(m_y,m_x),(n_y,n_x)}$ is the (horizontal) distance between the (m_y, m_x)-th and (n_y, n_x)-th grid points, and ℓ_{h} is the horizontal correlation radius, set to 200 km;
- $z_{\mathrm{bl}} = 5$ is the number of levels in the boundary layer, in which most of the chemistry processes happen;
- the standard deviation p is a parameter controlling the magnitude of the covariances, whose value must be determined.

When using the OI algorithm, the initial forecast estimate $\mathbf{x}_0^{\mathsf{f}}$ is taken as the state of the atmosphere predicted by the reference simulation of subsection 6.1.3 at the initial time t_0 of the DA experiment. In each analysis step, the $N_{\mathrm{y}} \times N_{\mathrm{y}}$ matrix $\mathbf{HBH}^{\mathsf{T}} + \mathbf{R}$ depends on the current set of available observations. It can be entirely computed because the number of observations N_y is moderate (a few hundreds) and because the observation operator $\mathbf{H}$ is sparse. The mean update is then computed using a Cholesky factorisation of the matrix $\mathbf{HBH}^\mathsf{T} + \mathbf{R}$. By construction of the background error covariance matrix $\mathbf{B}$, each analysis step only corrects the ozone concentration in the boundary layer. Finally, for simplicity the analysis error covariance matrix $\mathbf{P}^\mathsf{a}$ is not computed.

6.2.3 The LETKF algorithm

To implement the LETKF algorithm, we first need to specify the ensemble size N_e. State-of-the-art studies in atmospheric chemistry (see, *e.g.*, Gaubert 2013; Haussaire 2017, and references therein) tend to show that it is reasonable to use between 20 and 50 members in the ensemble. The benefit from using larger ensembles are moderate and do not compensate for the increase in computational cost, as long as localisation is used. Therefore, in our experiments we use an ensemble of $N_\mathrm{e} = 20$ members.

The initial forecast ensemble $\mathbf{E}_0^\mathsf{f}$ is constructed as a random draw from the distribution $\mathcal{N}[\bar{\mathbf{x}}_0^\mathsf{f}, \mathbf{B}]$, where $\bar{\mathbf{x}}_0^\mathsf{f}$ is the state of the atmosphere predicted by the reference simulation at the initial time t_0, and $\mathbf{B}$ is the background error covariance matrix, constructed exactly as in the previous subsection with a standard deviation p set to $32\,\mu\mathrm{g/m}^3$.

Preliminary experiments with the LETKF algorithm using both vertical and horizonal localisation (not illustrated here) have shown that the vertical localisation is unnecessary. Therefore, in our implementation of the LETKF algorithm, vertical localisation is disabled. As a result, the analysis step is split into $P_\mathrm{y} \times P_\mathrm{x}$ local analyses. During the (n_y, n_x)-th local analysis:

- the anomalies related to the q-th observation are tapered by a factor
$$\sqrt{G\left(\frac{d_{(n_y,n_x),q}}{\ell}\right)},$$
where G is the GC function introduced in subsection 2.5.3, $d_{(n_y,n_x),q}$ is the horizontal distance between the (n_y, n_x)-th grid point and the site of the q-th observation, and ℓ is the localisation radius;

- we update the $P_\mathrm{s} \times P_\mathrm{z}$ variables corresponding to the (n_y, n_x)-th grid point (one variable per chemical species and per vertical level).

As presented in subsection 2.5.2, in order to mitigate the sampling errors, additive inflation is used before the forecast step, which is equivalent to using an additional model error $\boldsymbol{e}^\mathsf{m}$ *before* integration. The model error is drawn from the distribution $\mathcal{N}[\mathbf{0}, \mathbf{Q}]$, where the model error covariance matrix $\mathbf{Q}$ is constructed exactly as the background error covariance matrix $\mathbf{B}$ from the previous subsection with a standard deviation p which needs to be specified. By contrast with the OI algorithm, even though $\mathbf{Q}$ only defines covariances between ozone concentrations, interspecies correlation can develop during the forecast step, which means that the analysis step can correct the concentration of all species at all vertical levels.

6.2.4 The LPF–X algorithms

The implementation of the two LPF–X algorithms is very similar to that of the LETKF algorithm described in the previous subsection: the ensemble size N_e is set to 20, the initial forecast ensemble $\mathbf{E}_0^\mathsf{f}$ is constructed using the same method, vertical localisation is disabled, and additive inflation is used before the forecast step.

Furthermore, the local importance weight vectors $\mathbf{w}_{N_\mathrm{b}:1}$ are computed using the Gaussian formulation, with equation (4.25), and there are $P_\mathrm{y} \times P_\mathrm{x}$ local blocks, each of them containing $P_\mathrm{s} \times P_\mathrm{z}$ variables (one variable per chemical species and per vertical level).

When using systematic resampling, the implementation of the local updates is straightforward. By contrast, when using anamorphosis, the implement of the local updates need to be described. For each variable $n \in (N_\mathrm{x} : 1)$, the transport map $\mathcal{T}_n$ is computed using the cdf of the n-th regularised marginal empirical forecast and analysis densities $\bar{\pi}_n^\mathsf{f}$ and $\bar{\pi}_n^\mathsf{a}$, as defined by equations (4.35) and (4.36). As for the experiments of chapter 5, the regularisation kernel $\mathcal{K}$ is chosen to be the Student's t-distribution with two degrees of freedom. However, we need to compute a local update for different chemical species, for which the concentration values may not have the same order of magnitude. Therefore, the resolution used to compute the transport map $\mathcal{T}_n$ is computed independently for each chemical species and each vertical level, by considering statistical properties (maximum, median) of the reference simulation. Finally, the forecast and analysis regularisation bandwidths h^f and h^a are both set to 0.25. This value has been deduced from preliminary experiments (not illustrated here).

6.2.5 The LPF–Y algorithm

The implementation of the LPF–Y algorithm is very similar to that of the other ensemble DA algorithms: the ensemble size N_e is set to 20, the initial forecast ensemble $\mathbf{E}_0^\mathsf{f}$ is constructed using the same method, vertical localisation is disabled, and additive inflation is used before the forecast step.

For this DA system, the second-order propagation method is implemented as follows. Suppose that we describe the assimilation of the q-th observation, and let ψ be the resampling map computed using the adjustment-minimising systematic resampling algorithm with the global importance weight vector $\mathbf{w}$. The update in observation space is computed as

$$\forall i \in (N_\mathrm{e} : 1), \quad \Delta y_q(i) \triangleq \mathbf{H}_q\mathbf{x}\big(\psi(i)\big) - \mathbf{H}_q\mathbf{x}(i). \tag{6.10}$$

The observation operator $\mathbf{H}_q$ being a bilinear observation operator, Δy_q can be interpreted as an update in state space, at a fictitious additional grid point which would lie at the q-th observation site. Therefore, it can be propagated to all variables in the V region using the method described in paragraph 4.3.3.3.

6.2.6 Short summary

The OI algorithm has two parameters: the observation standard deviation r, and the standard deviation p of the background error covariance matrix $\mathbf{B}$. Besides the ensemble size N_e, the four ensemble DA algorithms have three parameters: the observation standard deviation r, the standard deviation p of the model error covariance matrix $\mathbf{Q}$, and the localisation radius ℓ.

By contrast with the experiments of chapter 5, no post-regularisation step is used for the LPF algorithms. Indeed, in this DA system, the additive inflation method is enough to mitigate the sample impoverishment phenomenon. This is most probably related to the stability of the dynamical model $\mathcal{M}$.

6.3 Numerical experiments

6.3.1 The performance indicators

For the numerical experiments of this chapter, three different performance indicators are considered.

1. the average RMSE per station, which is defined as the average over all stations of the RMSE between the time series of predicted and measured ozone concentration at a station;

2. the average correlation per station, which is defined in a similar way as the average RMSE per station;

3. the instantaneous mean absolute error (MAE), which is defined as the average over all stations of the instantaneous absolute error between the predicted and measured ozone concentration at a station.

In every case, the indicator is restricted to a given group (`assim` or `valid`, as introduced in subsection 6.2.1) and computed using either the forecast estimate $\mathbf{x}^{\mathsf{f}}$ or the analysis estimate $\mathbf{x}^{\mathsf{a}}$.[2] Of course, these indicators are imperfect, because they consist in a comparison with noisy observations, but this is the best we can do given the fact that the true ozone concentration field is unknown. Finally, in order to relax the dependence to the initial ensemble, the first day of assimilation is removed from the time series.

6.3.2 Choosing the parameters

The OI algorithm has two parameters: the observation standard deviation r, and the standard deviation p of the background error covariance matrix $\mathbf{B}$. However, the analysis step of the OI algorithm relies on the computation of the Kalman gain matrix $\mathbf{K}$, defined as

$$\mathbf{K} = \mathbf{B}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{B}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}, \tag{6.11}$$

which actually only depends on the ratio p/r. Figure 6.7 shows the evolution of the average RMSE per station as a function of the ratio p/r for the OI algorithm. As expected, the RMSE score for the group `assim` (the group containing the observations which are assimilated) is monotonically decreasing as the ratio p/r increases. Indeed, when the observation standard deviation r is small (compared to the standard deviation p), the algorithm considers that the

[2] In particular, this means that the reported forecast indicators correspond to a 1 h forecast.

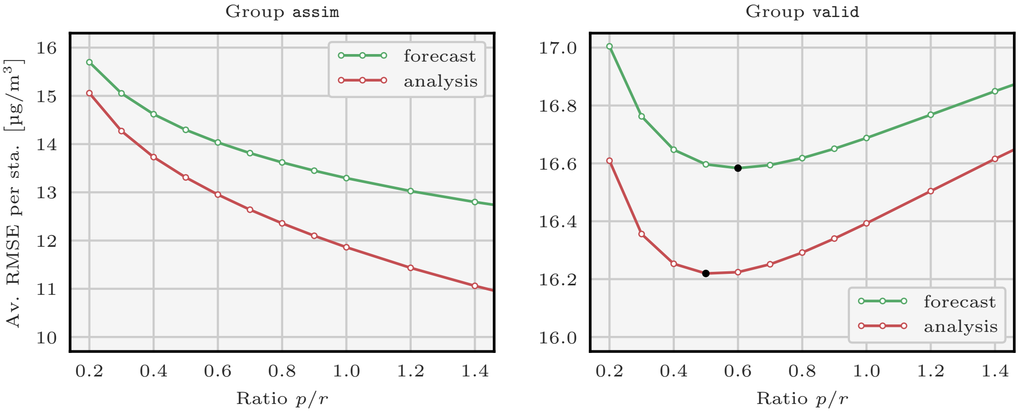

Figure 6.7: Evolution of the forecast (in green) and analysis (in red) average RMSE per station as a function of the ratio p/r for the OI algorithm. The indicators are computed for the groups `assim` (left panel) and `valid` (right panel). For the group `valid`, the optimal scores are shown with a black marker. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

observations are precise and adjusts the ozone concentrations to fit the observations as best as possible. However in this case, the statistical content of the observations is overestimated, as shown by the increase in RMSE score for the goup `valid` (the group containing the observations which are non assimilated and kept for cross-validation). This phenomenon is known as *overfitting*. Similar behaviour has been observed with all DA algorithms tested in this chapter. Therefore, from now on we focus on the indicators for the group `valid`.

Figure 6.8 shows the evolution of the average RMSE per station as a function of the ratio p/r for the LETKF algorithm. This evolution shows some similarity with that for the OI algorithm, yet some differences can be noticed. In this case, the scores do not only depend on the ratio p/r, as shown by the different evolutions when the observation standard deviation r is different. This was expected, because the standard deviation p of the model error covariance matrix $\mathbf{Q}$ influences the forecast sample covariance matrix $\bar{\mathbf{P}}^{\mathrm{f}}$ in a nonlinear fashion. Furthermore, the optimal ratio p/r is much larger when considering the forecast RMSE score than when considering the analysis RMSE score. In other words, for a fixed observation standard deviation r, the model perturbation (through the use of additive inflation) has to be larger when considering the forecast RMSE score than when considering the analysis RMSE score. This shows that there is room for improvement in the design of the dynamical model. Similar results have been observed for all ensemble DA algorithms tested in this chapter.

[3] In these experiments, it turns out that the value of ℓ which minimises the forecast RMSE score is approximately equal to the value of ℓ which minimises the analysis RMSE score.

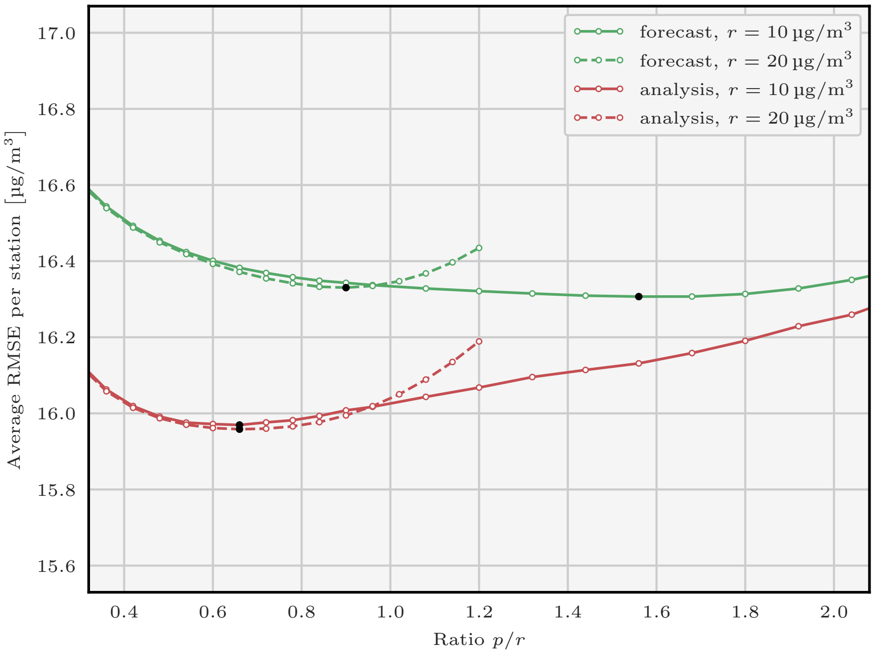

Figure 6.8: Evolution of the forecast (in green) and analysis (in red) average RMSE per station as a function of the ratio p/r for the LETKF algorithm. The observation standard deviation r is set either to $10\,\mu g/m^3$ (continuous lines) or or to $20\,\mu g/m^3$ (dashed lines), and for each value of the couple of parameters (r, p) the localisation radius ℓ is optimally tuned to yield the lowest RMSE score.[3] In each case, the optimal score is shown with a black marker. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

Table 6.4: Parametrisation of the DA algorithms for the experiments shown in subsection 6.3.3. For the OI, p is the standard deviation of the background error covariance matrix $\mathbf{B}$. For the ensemble DA algorithms, p is the standard deviation of the model error covariance matrix $\mathbf{Q}$.

Algorithm	Obs. std. dev. r [µg/m^3]	Std. dev. p [µg/m^3]	Loc. rad. ℓ [km]
OI	12	6.6	–
LETKF	12	7.2	220
LPF–X (sys.)	12	7.2	220
LPF–X (ana.)	12	7.2	220
LPF–Y	12	10.5	220

6.3.3 Optimal results

6.3.3.1 Approximate optimal parametrisations

In this section, we focus in parametrisations in which the analysis average RMSE per station is (close to) optimal. For the OI algorithm, the parametrisation is directly deduced from figure 6.7. For the ensemble DA algorithms, we found a common parametrisation which is approximately optimal for the LETKF algorithm, as well as for both tested LPF–X algorithms. Finally, for the LPF–Y algorithm, the standard deviation p of the model error covariance matrix $\mathbf{Q}$ had to be increased a bit to obtain a parametrisation close to optimal. This is summarised in table 6.4. In all cases, the observation standard deviation r is set to 12 µg/m^3, in agreement with most other studies (see, *e.g.*, Haussaire 2017, and references therein).

6.3.3.2 Single station trajectory

Figure 6.9 shows the time series of ozone concentration at the 10th station (located in Germany, see figure 6.1). With the reference simulation, the ozone concentration is:

1. highly underestimated in the morning, until 9:00 UTC;
2. a bit overestimated from 10:00 UTC to 18:00 UTC;
3. a bit underestimated in the evening, starting at 19:00 UTC.

In general, with DA the ozone concentration predicted using the analysis estimate $\mathbf{x}^{\mathsf{a}}$ is closer to the measurements than the ozone concentration predicted by the debiased reference simulation. More specifically:

1. the underestimation in the morning, though not entirely absorbed, is mitigated;
2. the ozone concentration predicted in the afternoon has the same order as the measured ozone concentration;
3. the underestimation in the evening is also mitigated (except when using the OI algorithm).

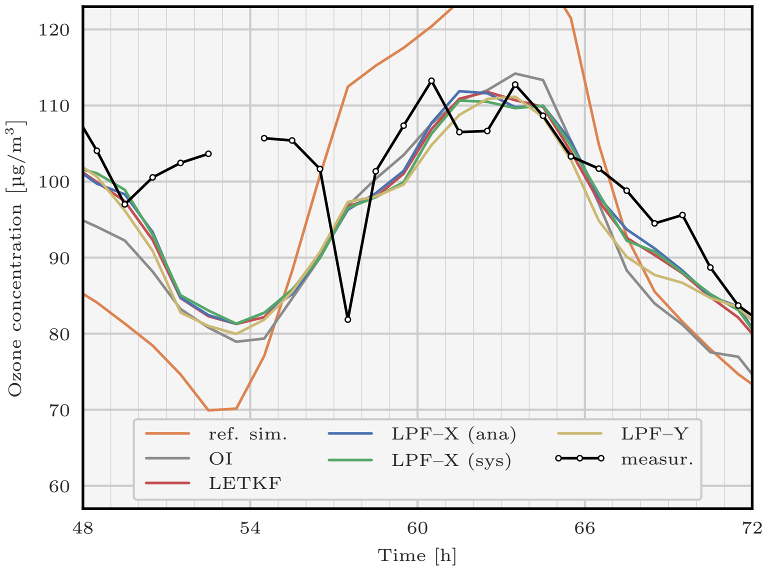

Figure 6.9: Time series of ozone concentration at the 10th station during the third day of assimilation for the measurements (in black), for the debiased reference simulation (in orange), and for the DA algorithms. For the DA algorithms, the prediction is computed from the analysis estimate $\mathbf{x}^{\mathrm{a}}$. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

Furthermore it should be noted that low ozone concentrations at 9:30 UTC and at 14:00 UTC are measured and not predicted at all. In summary, in this case using DA is always beneficial. Ensemble DA algorithms seem to have the edge over the OI algorithm. No clear ranking between ensemble DA algorithms emerge from these experiments.

6.3.3.3 Concentration maps

Figure 6.10 shows, at a specific time, the mean and the spread of the ozone concentration at ground level in the analysis ensemble $\mathbf{E}^{\mathrm{a}}$ for the LPF–X algorithm with anamorphosis. Furthermore, figures 6.11 and 6.12 shows, at the same time, the ozone concentration at ground level of two ensemble members in $\mathbf{E}^{\mathrm{a}}$ for the LPF–X algorithm with systematic resampling and with anamorphosis.

As expected, figure 6.10 shows that the spread of $\mathbf{E}^{\mathrm{a}}$ is large in the regions without observations (typically in the Mediterranean and North seas) and small in the densely observed regions (typically in central Europe). Figure 6.11 shows that the LPF–X algorithm with systematic resampling is characterised by noticeable spatial discontinuities in $\mathbf{E}^{\mathrm{a}}$. At this point, it is not clear whether these discontinuities impact the efficiency of the assimilation, but it is satisfactory to see, in figure 6.12, that the discontinuities have been mitigated when using the LPF–X algorithm with anamorphosis.[4] This confirms the general conclusion of

[4] The effect described here is not intense and may not be obvious in printed versions of the maps.

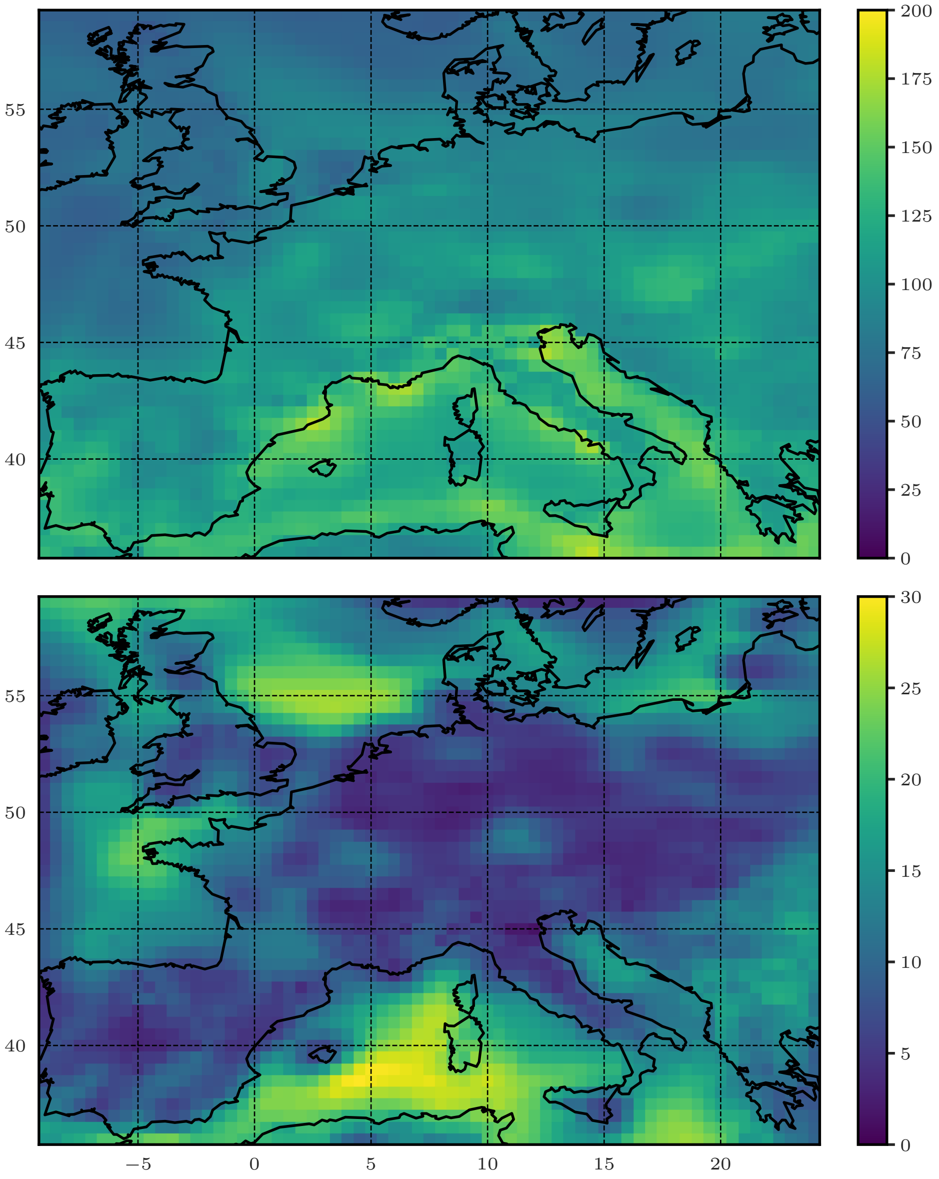

Figure 6.10: Mean (top panel) and spread (bottom panel) of the ozone concentration at ground level in the analysis ensemble $\mathbf{E}^a$ for the LPF–X algorithm with anamorphosis at 12:30 the fifth day of assimilation. The values are given in µg/m^3. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

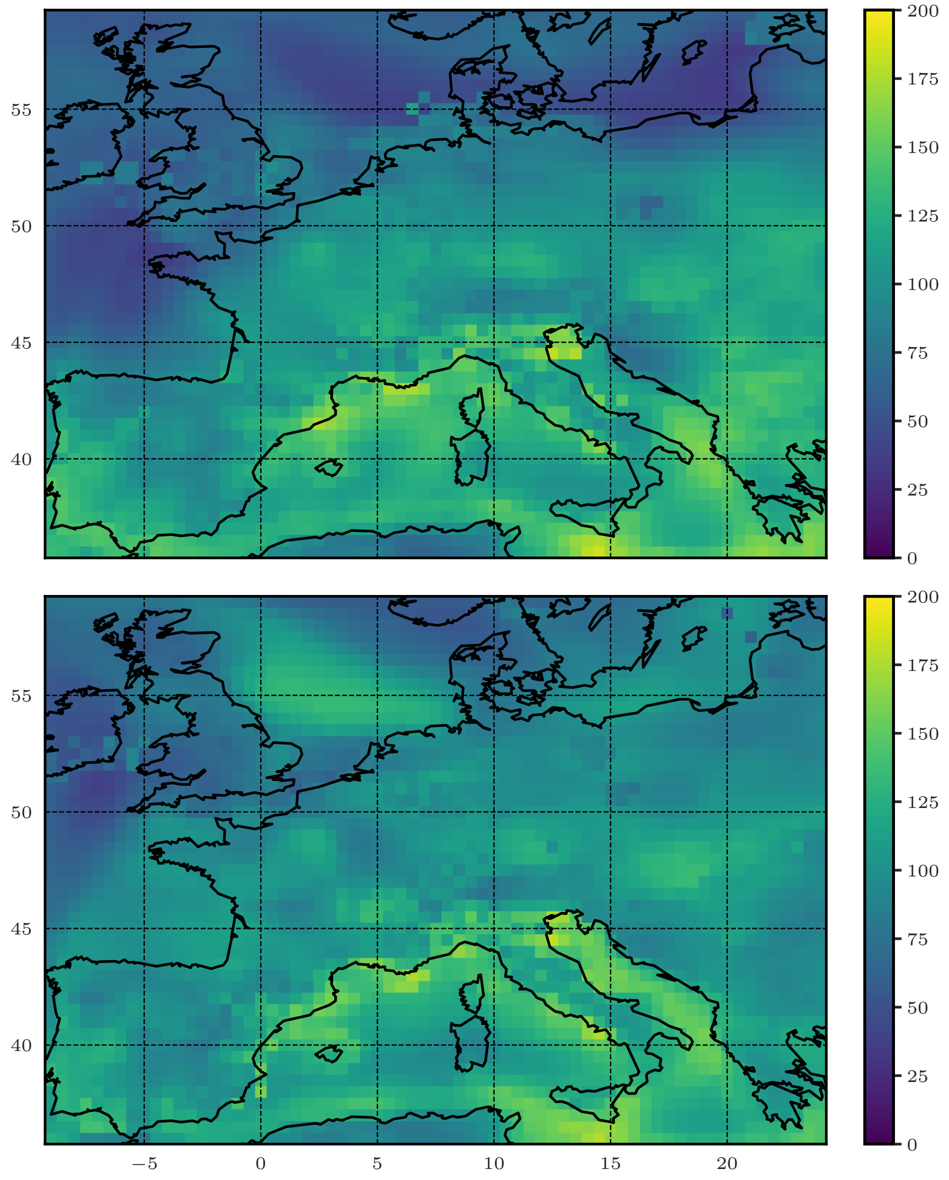

Figure 6.11: Ozone concentration at ground level of two ensemble members in the analysis ensemble $\mathbf{E}^{\mathrm{a}}$ for the LPF–X algorithm with systematic resampling at 12:30 the fifth day of assimilation. The values are given in µg/m^3. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

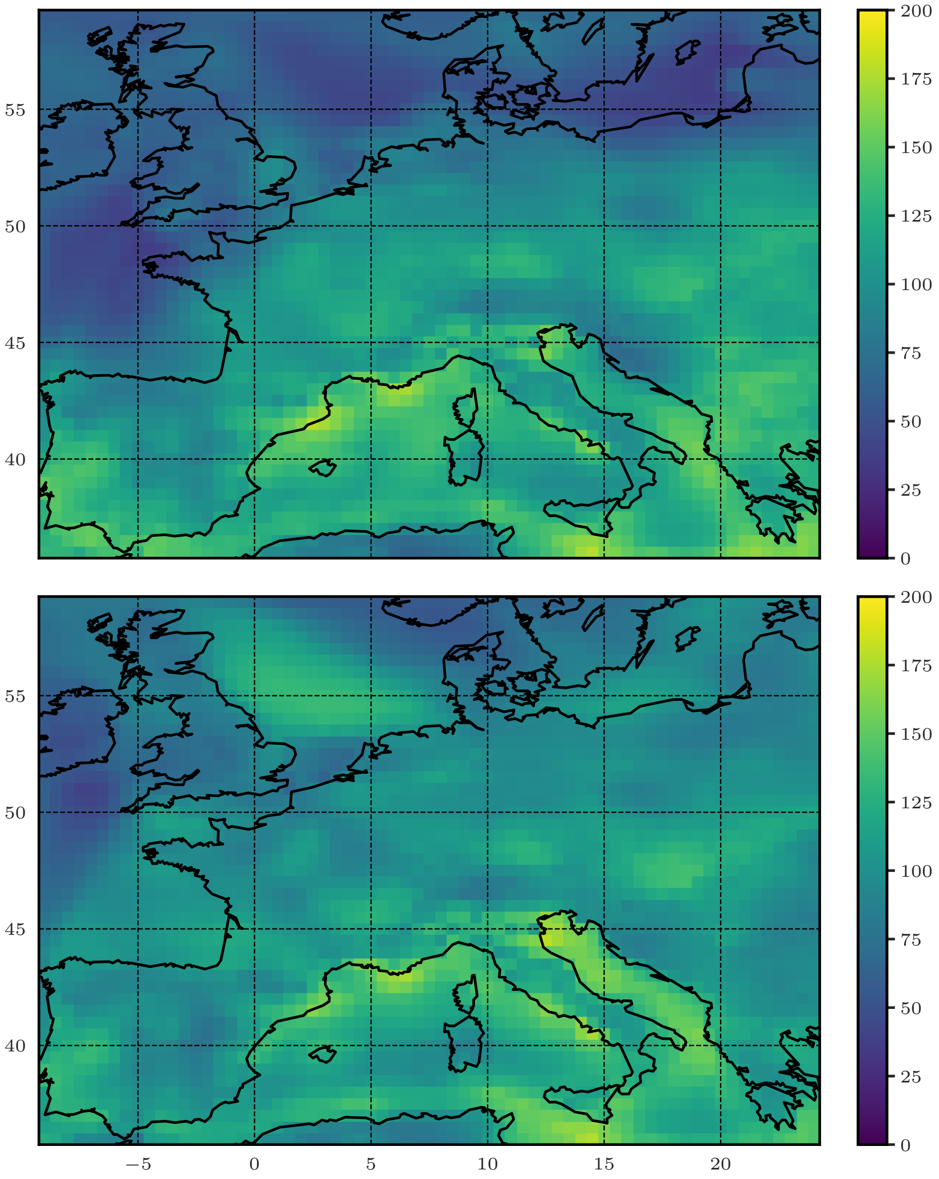

Figure 6.12: Ozone concentration at ground level of two ensemble members in the analysis ensemble $\mathbf{E}^a$ for the LPF–X algorithm with anamorphosis at 12:30 the fifth day of assimilation. The values are given in µg/m^3. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

Table 6.5: Averaged results for the debiased reference simulation and for the DA algorithms. For the debiased reference simulation, the average correlation per station is 66 %, whereas for all DA algorithms, the average (forecast and analysis) correlation per station is between 76 % and 78 %. In all cases, the reported wall-clock time is the total wall-clock time (that is, for 119 forecast steps and for 120 analysis steps). For the debiased reference simulation and for the OI algorithm, parallelisation is enabled in the `Polair3DChemistry` model using 20 `OpenMP` threads. For the ensemble DA algorithms, parallelisation is enabled in the independent forecast of the 20 members of the ensemble using 20 `OpenMP` threads. Finally, for the LETKF and the LPF–X algorithms, parallelisation is enabled in the $P_\mathrm{y} \times P_\mathrm{x} = 3149$ independent local updates using 20 `OpenMP` threads.

Algorithm	Average RMSE per station $[\mu\mathrm{g/m}^3]$		Wall-clock time $[\mathrm{s}]$	
	Forecast (1 h)	Analysis	Forecast (1 h)	Analysis
ref. sim.	19.692	19.692	428.0	–
OI	16.587	16.219	453.8	92.0
LETKF	16.389 ± 0.007	15.967 ± 0.010	2270.8	41.7
LPF–X (sys.)	16.440 ± 0.022	16.007 ± 0.027	2312.3	24.0
LPF–X (ana.)	16.411 ± 0.017	16.000 ± 0.017	2295.6	119.2
LPF–Y	16.503 ± 0.021	16.103 ± 0.022	2264.5	308.0

chapter 5: using optimal transport for the local updates in LPF–X algorithms is an efficient way of mitigating the unphysical discontinuities.

Finally, we conclude this preliminary discussion by mentioning the fact that the corresponding maps for the LETKF and LPF–Y algorithms (not illustrated here) are visually very similar to the maps for the LPF–X algorithm with anamorphosis at one exception: the spread of $\mathbf{E}^\mathrm{a}$ is larger for the LPF–Y algorithm than for the other ensemble DA algorithms in the regions without observations. This can be explained by the fact that the LPF–Y algorithm uses a larger standard deviation p for the model error covariance matrix $\mathbf{Q}$ ($10.5\,\mu\mathrm{g/m}^3$ versus $7.2\,\mu\mathrm{g/m}^3$ for the other ensemble DA algorithms).

6.3.3.4 Average results

Using the parametrisations described in paragraph 6.3.3.1, each DA experiment[5] is performed again 10 times, each using a different random seed, on the same computational platform with 16 cores. The results, averaged over the 10 realisations, are reported in table 6.5. Furthermore, figure 6.13 shows the average daily cycle of instantaneous analysis MAE, once again averaged over the 10 realisations.

From the statistical indicators, it is clear that using DA in this system is beneficial. For example, the improvement in average RMSE per station is about 19 % for the analysis and about 17 % for the 1 h forecast. The performance of all five DA algorithms, as measured by the average RMSE and correlation per station as well as by the instantaneous MAE, are almost equivalent. In all cases, the ensemble DA algorithms seems to have the edge over the

[5] Except when using the fully deterministic OI algorithm.

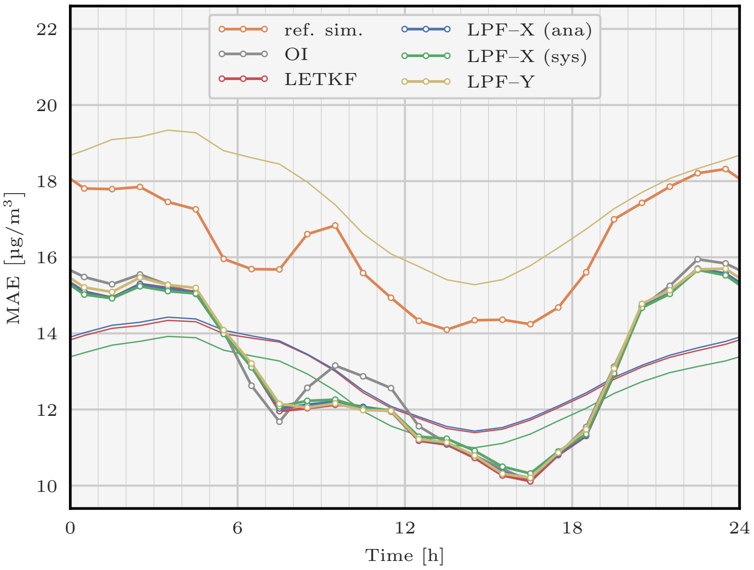

Figure 6.13: Average daily cycle of instantaneous analysis MAE for the debiased reference simulation (in orange), and for the DA algorithms. For each ensemble DA algorithm, the average daily cycle of average ensemble spread in ozone concentration at ground level is shown with a thin line of the same color but without markers. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

OI algorithm. However, it is not clear whether the small gain in average RMSE per station (about 1 %) is sufficient to justify the huge increase in forecast wall-clock time due to the use of an ensemble of $N_\mathrm{e} = 20$ members.

The scores for the LPF algorithms are very similar to those for the LETKF algorithm, which is a première in atmospheric chemistry. The LPF–X algorithm with systematic resampling, in spite of the discontinuity issues illustrated in paragraph 6.3.3.3, yields similar average (forecast and analysis) RMSE per station and instantaneous (analysis) MAE scores as the LETKF algorithm, while being about 40 % faster in the analysis step. The LPF–X algorithm with anamorphosis also yields similar scores as the LETKF algorithm, but it is about 185 % slower in the analysis step. This is explained by the fact that it is numerically demanding to compute, for each local analysis, an anamorphosis transformation for every species and every level (that is, a total of $P_\mathrm{s} \times P_\mathrm{z} = 572$ transformations to compute), instead of computing a single transformation matrix $\mathbf{T}_\mathrm{e}$ in ensemble space. Finally, the average (forecast and analysis) RMSE per station is slightly higher for the LPF–Y algorithm. Furthermore, for this algorithm, the average ensemble spread in ozone concentration at ground level is much higher than for all other DA algorithm. This is explained by the fact that the LPF–Y algorithm uses a large standard deviation p for the model error covariance matrix $\mathbf{Q}$.

Finally, from the time series of instantaneous analysis MAE, we see that all DA algorithms yield better performances during the day, when the ozone concentrations are, on average, at their highest levels. This pattern can be seen in the debiased reference simulation as well. Therefore, we conclude that the performance of the DA algorithms is highly impacted by the lower accuracy of the dynamical model $\mathcal{M}$ during the night, which is a common characteristic of all CTMs.

6.3.3.5 Rank histograms

As a complement to the average results presented in the previous paragraph, we compute rank histograms for the ensemble DA algorithms. At each station, the histogram is obtained by computing the rank of the measured ozone concentration in the ensemble of predicted ozone concentrations as determined by the analysis ensemble $\mathbf{E}^\mathsf{a}$. The mean histograms, averaged over all stations and over the 10 realisations, are reported in figure 6.14. All algorithms are characterised by U-shaped histograms. This shows that, in this case, DA is indeed an efficient way of filtering the observation noise. Furthermore, the histograms are all characterised by a slight but noticeable negative bias.

6.3.3.6 Non-Gaussian diagnostic

As explained in subsection 6.2.1, the dynamical model $\mathcal{M}$ is expected to be strongly nonlinear as a result of the chemical processes, which would result in a non-Gaussian forecast distribution ν^f. Yet, there is little difference between the scores of the LETKF algorithms and of the LPF algorithms. Therefore, in this section we want to measure the deviation from Gaussianity in the numerical experiments with ensemble DA algorithms. To do this, we use the statistical properties of the ensemble.

For each algorithm, one out of the 10 realisations is selected, and the first day of assimilation is removed from the time series. For each each variable in the ozone concentration field at

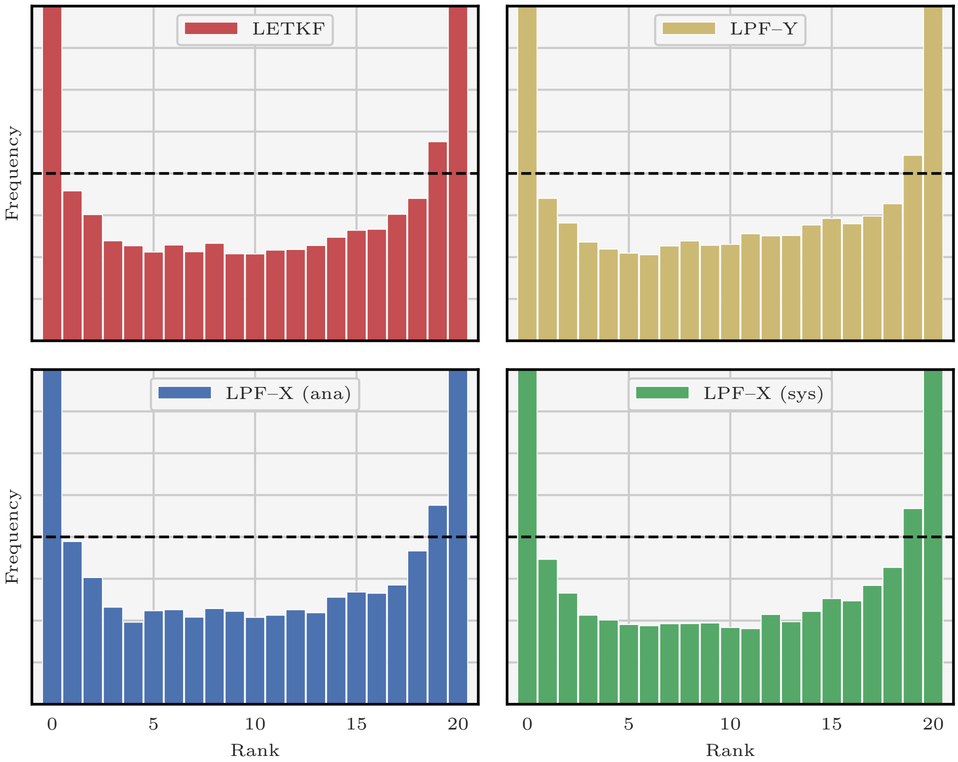

Figure 6.14: Rank histograms for the ensemble DA algorithms. A dashed black line indicates the ideal frequency, $(N_\mathrm{e}+1)^{-1}$. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

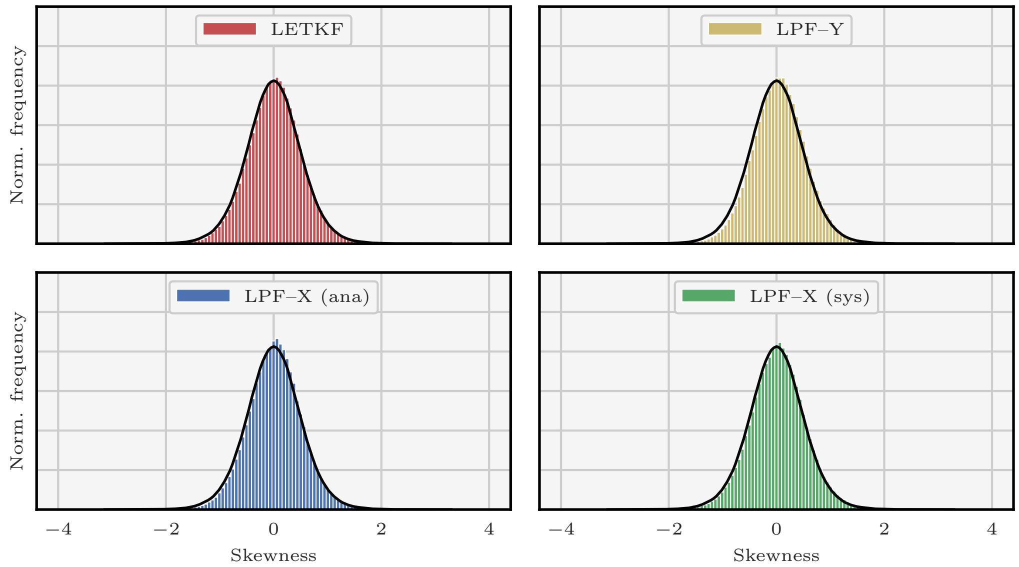

Figure 6.15: Empirical distribution of skewness in the normalised forecast ensemble $\mathbf{E}^{\mathrm{f}}$ for the ensemble DA algorithm. A black line indicates the empirical distribution of skewness for a random variable drawn from the distribution $\mathcal{N}[0,1]$. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

ground level, at each time step, we compute the skewness and the excess kurtosis[6] of the normalised forecast ensemble $\mathbf{E}^{\mathrm{f}}$ (in other words, the rescaled forecast ensemble $\mathbf{E}^{\mathrm{f}}$ with zeros mean and with a unit standard deviation). For comparison, we also computed the skewness and kurtosis of a normalised ensemble obtained by random draws from the distribution $\mathcal{N}[0,1]$. Figures 6.15 and 6.16 show the empirical distribution of these $4 \times 24 \times P_{\mathrm{y}} \times P_{\mathrm{x}} = 302\,304$ values of skewness and kurtosis for both the ensemble DA algorithms and the distribution $\mathcal{N}[0,1]$.

From this global, univariate point of view, using an ensemble of $N_{\mathrm{e}} = 20$ members is not enough to distinguish non-Gaussian effect in $\mathbf{E}^{\mathrm{f}}$. This means that, contrary to what we expected, these DA experiments, with a time interval between consecutive observation Δt equal to 1 h and with a significant debiasing, put the system in a quasi linear dynamical regime for which there might not be any particular advantage of using PF over EnKF algorithms.

6.3.4 Complementary experiments

As a complement to the optimal results presented in subsection 6.3.3, we present here two series of experiments. In the first set of experiments, the ensemble size is increased from $N_{\mathrm{e}} = 20$ members to $N_{\mathrm{e}} = 40$ members. In the second set of experiments, the input parameters of the `Polair3DChemistry` model are perturbed.

[6] Simply called the kurtosis in the following.

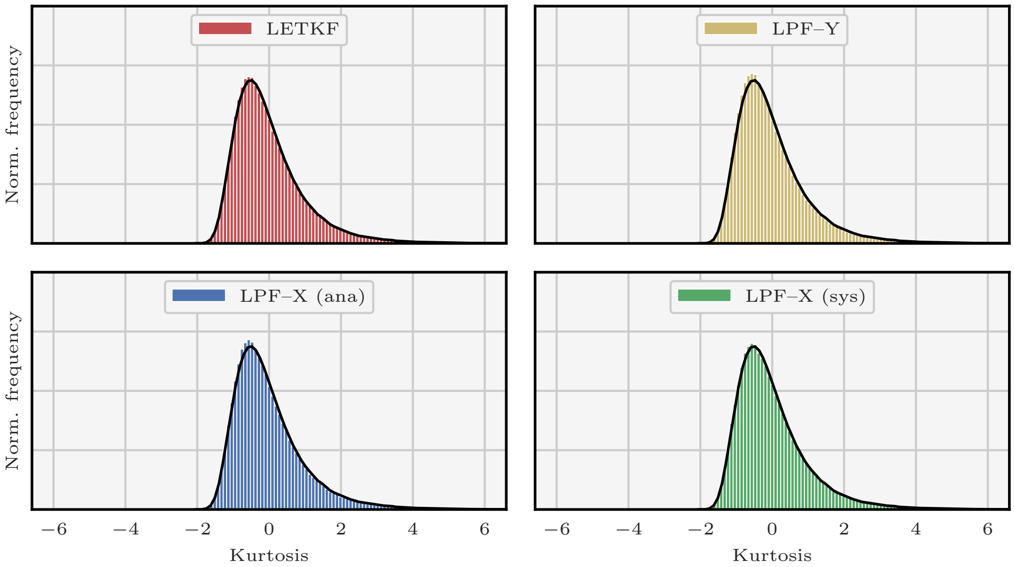

Figure 6.16: Empirical distribution of kurtosis in the normalised forecast ensemble $\mathbf{E}^{\mathrm{f}}$ for the ensemble DA algorithm. A black line indicates the empirical distribution of kurtosis for a random variable drawn from the distribution $\mathcal{N}[0,1]$. The DA system is the `Polair3D-Chemistry` model with the observations from `Airbase`.

6.3.4.1 Increasing the ensemble size

Without modification in the parametrisation described in table 6.4, the DA experiments are performed again 10 times for the LETKF algorithm and the LPF–X algorithm with anamorphosis, using an ensemble of $N_{\mathrm{e}} = 40$ members instead of 20. The computational platform for these experiments is the same as for the experiments described in subsection 6.3.3. The results, averaged over the 10 realisations, are reported in table 6.6. Furthermore, figure 6.17 shows the average daily cycle of instantaneous analysis MAE, once again averaged over the 10 realisations.

There are only two significant differences between the case $N_{\mathrm{e}} = 20$ and $N_{\mathrm{e}} = 40$: the average ensemble spread in ozone concentration at ground level is slightly higher when using $N_{\mathrm{e}} = 40$ members, and the wall-clock time has increased. Of course, the parametrisation of the DA algorithms has been chosen for the case $N_{\mathrm{e}} = 20$, but it is still disappointing to note that average RMSE per station has been barely improved in the case $N_{\mathrm{e}} = 40$. Furthermore, we have checked that changing the parametrisation (not illustrated here) leads to the same conclusion: for the DA algorithms tested here, it is better, in terms of efficiency, to use an ensemble of $N_{\mathrm{e}} = 20$ members than $N_{\mathrm{e}} = 40$ members.

6.3.4.2 Perturbation of the input data

In the experiments presented so far, when using an ensemble DA algorithm, the collapse of the ensemble $\mathbf{E}$ is mitigated through the use of an additional model error $\boldsymbol{e}^{\mathrm{m}}$, and the standard

Table 6.6: Averaged results for the LETKF algorithm and the LPF–X algorithm with anamorphosis using an ensemble of $N_\mathrm{e} = 40$ members. For comparison, the results using an ensemble of $N_\mathrm{e} = 20$ members, reported in table 6.5, are reported again. In all cases, the average (forecast and analysis) correlation per station is between 76 % and 78 %. Furthermore, the reported wall-clock time is the total wall-clock time (that is, for 119 forecast steps and for 120 analysis steps). Parallelisation is enabled in the independent forecast of the N_e members of the ensemble using N_e `OpenMP` threads. Finally, parallelisation is enabled in the $P_\mathrm{y} \times P_\mathrm{x} = 3149$ independent local updates using 20 `OpenMP` threads.

Algorithm	Average RMSE per station $[\mu\mathrm{g/m}^3]$		Wall-clock time $[\mathrm{s}]$	
	Forecast (1 h)	Analysis	Forecast (1 h)	Analysis
LETKF, $N_\mathrm{e} = 20$	16.389 ± 0.007	15.967 ± 0.010	2270.8	41.7
LETKF, $N_\mathrm{e} = 40$	16.380 ± 0.009	15.960 ± 0.010	3268.0	93.9
LPF–X, $N_\mathrm{e} = 20$	16.411 ± 0.017	16.000 ± 0.017	2295.6	119.2
LPF–X, $N_\mathrm{e} = 40$	16.394 ± 0.009	15.981 ± 0.010	3298.1	405.5

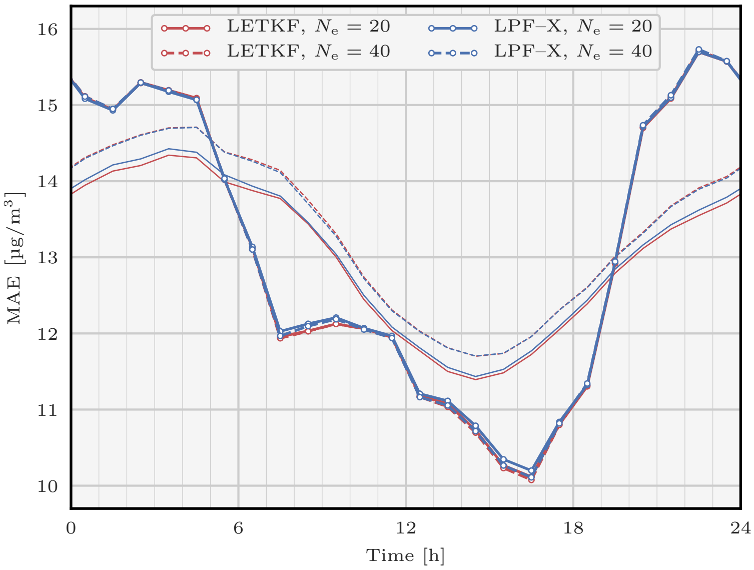

Figure 6.17: Average daily cycle of instantaneous analysis MAE for the LETKF algorithm (in red) and the LPF–X algorithm with anamorphosis (in blue). The ensemble size is either $N_\mathrm{e} = 20$ (continuous lines) or $N_\mathrm{e} = 40$ (dashed lines). In each case, the average daily cycle of average ensemble spread in ozone concentration at ground level is shown with a thin line without markers. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

deviation p of the model error covariance matrix $\mathbf{Q}$ is adjusted in such a way that the spread of the forecast ensemble $\mathbf{E}^{\mathsf{f}}$ is sufficient. As mentioned in subsection 6.1.3, an important source of uncertainty in the reference simulation is the potential bias in the input database. Therefore, an alternative to additional model error is to use input data perturbations, in other words introducing perturbations in the input database. Such perturbations must be constructed in line with the uncertainty in the input data, and the imbalance in the forecast ensemble $\mathbf{E}^{\mathsf{f}}$ is expected to be less critical than with additional model error $\boldsymbol{e}^{\mathsf{m}}$.

Different methods can be used to design the input data perturbations. Taking inspiration from the methods described by Wu et al. (2008) and Boynard et al. (2011), we chose to implement the input data perturbations as follows. For each input field to perturb, at each integration time step, a set of N_{e} independent *multiplicative* perturbation fields is drawn from the distribution $\mathcal{LN}[-\mu\mathbf{I}, q\mathbf{I}]$, where the parameter q is to be determined, and where the parameter μ is chosen in such a way that the expected value of the perturbation field is null: $\mu = q^2$. Horizontal and vertical correlations (when relevant) are then applied to each perturbation field using the same method as for the additional model error $\boldsymbol{e}^{\mathsf{m}}$, described in subsection 6.2.3, but with potentially different correlation radii. In order to avoid a compensation of the perturbation fields, a temporal autocorrelation is applied using the following method. Let $\mathbf{v}_i(t)$ be the vector containing the perturbation field applied at time t to a given input field. The perturbation field $\mathbf{v}_i(t+\delta t)$ applied at time $t+\delta t$ to the same input field is given by

$$\mathbf{v}_i(t+\delta t) = \alpha\,\mathbf{v}_i(t) + \sqrt{1-\alpha^2}\,\mathbf{z}_i, \tag{6.12}$$

where $\mathbf{z}_i$ is the correlated random draw obtained as described above, and $\alpha = 1 - \delta t/\tau$ with τ being the autocorrelation time. The parametrisation of the input data perturbations used in this paragraph is described in table 6.7.

Preliminary experiments using this parametrisation of the input data perturbations (not illustrated here) have shown that additional model error $\boldsymbol{e}^{\mathsf{m}}$ is still mandatory to mitigate the collapse of the ensemble $\mathbf{E}$. Therefore, the DA experiments are performed again 10 times for the LETKF algorithm and the LPF–X algorithm with anamorphosis using the input data perturbations described in table 6.7, but without further modification to the parametrisation described in table 6.4. The computational platform for these experiments is the same as for the experiments described in subsection 6.3.3. The results, averaged over the 10 realisations, are reported in table 6.8. Furthermore, figure 6.18 shows the average daily cycle of instantaneous analysis MAE, once again averaged over the 10 realisations.

By contrast with the experiments shown in the previous paragraph, the reduction in average analysis RMSE per station between the original LETKF algorithm and the LETKF algorithm with input data perturbations is about 20 % of the reduction in average analysis RMSE per station between the OI algorithm and the original LETKF algorithm. From figure 6.18, we note a small decrease in instantaneous MAE during the day, but also significant modifications in the average ensemble spread in ozone concentration at ground level. During the day, the average spread has increased, but during the night it has unexpectedly decreased. Finally, we note a significant increase in computational time, which corresponds to the computation of all perturbation fields. Again, the parametrisation of the DA algorithms has been chosen for the case without input data perturbation, however the results obtained in these experiments confirm the potential of the method. Several improvements in the design of the method can

Table 6.7: Parametrisation for the perturbations of the input data of the `Polair3D-Chemistry` model. For the boundary conditions, we only perturb the ozone, the nitrogen oxide, and the nitrogen dioxide fields, using the same perturbation field for all three species. For the emissions, the same perturbation field is used to perturb the surface emissions and the volume emissions at all levels for all emitted species. In all cases, the shape of the perturbed input field is reported in the second column. The horizontal and vertical correlation radii ℓ_h and ℓ_v are reported in the third and fourth columns. The standard deviation q of the perturbation distribution $\mathcal{LN}[-\mu\mathbf{I}, q\mathbf{I}]$ is reported in the fifth column. Finally, the autocorrelation time τ is reported in the last column.

Input field	Shape	ℓ_h	ℓ_v	q	τ
		[km]	[m]		[h]
Boundary conditions (z)	$P_\mathrm{y} \times P_\mathrm{x}$	800	–	0.30	6
Boundary conditions (y)	$P_\mathrm{z} \times P_\mathrm{x}$	500	3000	0.30	6
Boundary conditions (x)	$P_\mathrm{z} \times P_\mathrm{y}$	500	3000	0.30	6
Deposition velocity	$P_\mathrm{y} \times P_\mathrm{x}$	4000	–	0.25	6
Attenuation	$P_\mathrm{y} \times P_\mathrm{x}$	800	–	0.50	6
Emissions	$P_\mathrm{y} \times P_\mathrm{x}$	800	–	0.50	6
Vertical diffusion velocity	P_z	–	3000	0.40	6

Table 6.8: Averaged results for the LETKF algorithm and the LPF–X algorithm with anamorphosis using the input data perturbations described in table 6.7. For comparison, the results without input data perturbation, reported in table 6.5, are reported again. In all cases, the average (forecast and analysis) correlation per station is between 76 % and 78 %. Furthermore, the reported wall-clock time is the total wall-clock time (that is, for 119 forecast steps and for 120 analysis steps). Parallelisation is enabled in the independent forecast of the 20 members of the ensemble using 20 `OpenMP` threads. Finally, parallelisation is enabled in the $P_\mathrm{y} \times P_\mathrm{x} = 3149$ independent local updates using 20 `OpenMP` threads.

Algorithm	Average RMSE per station [µg/m^3]		Wall-clock time [s]	
	Forecast (1 h)	Analysis	Forecast (1 h)	Analysis
LETKF, no pert.	16.389 ± 0.007	15.967 ± 0.010	2270.8	41.7
LETKF, with pert.	16.331 ± 0.022	15.923 ± 0.017	3069.4	41.6
LPF–X, no pert.	16.411 ± 0.017	16.000 ± 0.017	2295.6	119.2
LPF–X, with pert.	16.368 ± 0.024	15.977 ± 0.020	3079.8	177.3

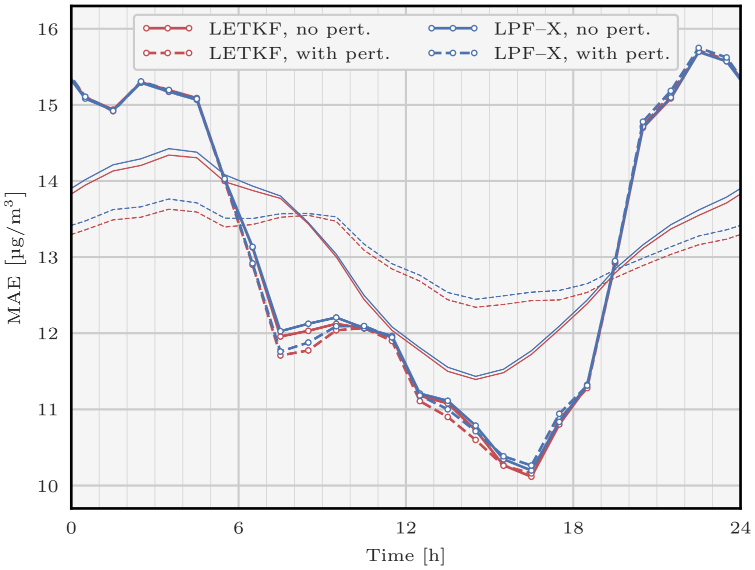

Figure 6.18: Average daily cycle of instantaneous analysis MAE for the LETKF algorithm (in red) and the LPF–X algorithm with anamorphosis (in blue). The ensemble size is either $N_\mathrm{e} = 20$ (continuous lines) or $N_\mathrm{e} = 40$ (dashed lines). In each case, the average daily cycle of average ensemble spread in ozone concentration at ground level is shown with a thin line without markers. The DA system is the `Polair3DChemistry` model with the observations from `Airbase`.

be conceived.

First, we have chosen, for the perturbations of the boundary conditions and of the emissions, to use the same perturbation fields for each perturbed chemical species, because it is convenient from a technical point of view. Yet, in order to avoid compensations in the perturbation fields, each chemical species should be independently perturbed. However, this would result in a significant increase in computational time, because the number of perturbation fields to compute would be increased by a factor $P_{\mathrm{s}} = 52$. A more affordable approach could be to separate the chemical species into four categories: (i) the ozone, (ii) the nitrogen oxides, (iii) the volatile organic compounds, and (iv) the remaining species. A perturbation field could then be computed for each category. Of course, if the increase in computational time is still too high, one can consider updating the perturbation fields once every other integration step instead of once every integration step.

Second, the input data perturbations could be designed to increase the average spread during the night to reflect the uncertainty in the chemical processes. During the day, the uncertainty in the chemical processes is reflected in the perturbation of the attenuation field, which in turn affects the photolysis. During the night, there is no photolysis, which means that perturbing the attenuation field is ineffective. From a technical point of view it is difficult to directly modify the rate of the chemical reactions in the `Polair3DChemistry` model. Therefore, an alternative, *ad hoc* approach could be to perturb the temperature field, which in turn affects the rate of the thermal chemical reactions.

6.4 Summary and discussion

In this chapter, we have described step-by-step the application of several DA algorithms to a realistic DA system. We have chosen the case study of the prediction of the tropospheric ozone concentration in western Europe during the summer 2009. Measurements of ozone concentration are taken from `Airbase`. They are available at several hundreds of stations, with a time interval between consecutive observations of 1 h. For our experiments, we have chosen to use the CTM `Polair3DChemistry` from the `Polyphemus` framework.

Using this model, we have constructed a highly biased reference simulation. A simple debiasing method, with 2 bias parameters per station in addition to 24 common bias parameters, is proposed and tested. With the debiasing, the reference simulation yields statistical indicators of the same order as typical CTMs.

We have then explained how to implement five DA algorithms: the OI algorithm, the LETKF algorithm, the LPF–X algorithm with systematic resampling, the LPF–X algorithm with anamorphosis, and the LPF–Y algorithm. In particular, for the ensemble DA algorithms, we have explained the implementation of additional model error. For the DA experiments, the stations have been divided into two groups. The first group is used during the analysis step, while the second group is kept for cross validation.

The results show that DA is effective in this system, with an improvement between 15 and 20 % in the average RMSE per station. The scores for the LPF algorithms are very similar to those for the LETKF algorithm, which is a première in atmospheric chemistry. In all cases, the ensemble DA algorithms seems to have the edge over the OI algorithm, however it is not clear whether the small gain in average RMSE per station (about 1 %) is sufficient to justify the huge increase in forecast wall-clock time due to the use of an ensemble of 20 members. For these experiments, we have shown that the deviation from Gaussianity in the forecast ensemble is hardly noticeable. Therefore, contrary to what we expected, these DA experiments, with a time interval between consecutive observation of 1 h and with a significant debiasing, put the system in a quasi linear dynamical regime for which there might not be any particular advantage of using PF over EnKF algorithms.

In complementary experiments, we have shown that a promising approach to further improve the performance of the ensemble DA algorithms is to use input data perturbations. However, further work is needed to fix the design of the input data perturbations. Finally, `Airbase` also provides measurements of other chemical species, and in particular of nitrogen dioxide. From a theoretical point of view, it would be desirable to implement multi-species assimilation. From a technical point of view, this is delicate, because it means that we would have to precisely control the uncertainty of all assimilated species. Furthermore, in this case, we would have to provide a composite score mixing all assimilated species, which would be non-trivial to define.

Part III

Covariance localisation in the ensemble Kalman filter

7 Implementing covariance localisation using augmented ensembles

Contents

In the EnKF, two localisation methods have emerged: DL (Houtekamer and Mitchell 2001; Ott et al. 2004), and CL (Hamill et al. 2001). DL consists of a collection of local and independent ensemble updates. This leads to efficient data assimilation algorithms, for example the LETKF algorithm. When using DL however, satellite observations cannot be assimilated without *ad hoc* approximations. By contrast, CL consists of a single ensemble update using a localised forecast sample covariance matrix. This is in practice much less simple to implement in a deterministic context, but it can be used to assimilate satellite observations without further approximations. The huge increase of satellite observations in the recent years justify the need for efficient implementations of CL in the EnKF.

EnKF algorithms using DL have been adapted to the case of satellite radiances (see, *e.g.*, Fertig et al. 2007; Miyoshi and Sato 2007). In these algorithms, the shape of the weighting function associated to a specific satellite channel is used to give an approximate location to this channel (usually the function mode). However, using a realistic one-dimensional model with satellite radiances, Campbell et al. (2010) have shown that this approach systematically yields higher errors than using CL.

In this chapter, following the work published in Farchi and Bocquet (2019), we focus on efficient implementations of CL in the deterministic EnKF. In this context, the literature shows a growing interest in using an augmented ensemble during the analysis step, that is when the ensemble size during the analysis step is larger than during the forecast step. In this case, the augmented ensemble size should be chosen in such a way that the augmented ensemble is large enough to accurately represent the forecast error covariance matrix. Buehner (2005) has proposed a method to construct a modulated ensemble which follows the localised forecast covariance based on a factorisation property shown by Lorenc (2003). This method has then been leveraged upon by Bishop and Hodyss (2009) and used in the literature to perform CL (Brankart et al. 2011; Bishop and Hodyss 2011; Leng et al. 2013; Bocquet 2016; Bishop et al. 2017). With an alternative point of view, Kretschmer et al. (2015) have included localisation in the ETKF algorithm by using a climatologically augmented ensemble. Finally, Lorenc (2017) has shown that the forecast error covariance matrix can be improved in hybrid ensemble variational DA systems by using time-lagged and time-shifted perturbations. Section 7.1 shows how CL can be implemented in the EnKF, in particular when using an augmented ensemble. In section 7.2, we describe in details how the augmented ensemble can be constructed. In section 7.3, the algorithms are tested using twin experiments of low-order one-dimensional models. In section 7.4, we explain how the methods can be used to assimilate satellite radiances, and we test the resulting algorithm using twin experiments of a multilayer extension of the L96 model. Finally, conclusions are given in section 7.5. In this chapter, unless specified otherwise, the DA system is the GL system. For simplicity, the time subscript k is systematically dropped in the equations.

7.1 Implementing CL in deterministic EnKF algorithms

7.1.1 The local ensemble square root Kalman filter

As presented in subsection 2.2.6, the analysis step of deterministic EnKF algorithms is split into two parts: first the mean update, and then the perturbation update. The mean update relies on the Kalman gain matrix $\mathbf{K}$ to yield the analysis estimate $\mathbf{x}^{\mathsf{a}}$, and it is implemented with equations (2.10a)–(2.10b). By contrast, the perturbation update relies on a square root formula to yield the analysis perturbations $\mathbf{X}^{\mathsf{a}}$. Without localisation, the perturbation update can take two different forms. It can be implemented either with equations (2.22a)–(2.22b), using a transformation matrix $\mathbf{T}_{\mathrm{e}}$ formulated in ensemble space, or with equations (2.23a)–(2.23b), using a transformation matrix $\mathbf{T}_{\mathrm{x}}$ formulated in state space.

As presented in subsection 2.5.3, when using CL, the forecast sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$ is replaced by its localised version $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$. By construction, the localisation matrix $\boldsymbol{\rho}\in\mathbb{R}^{N_{\mathrm{x}}\times N_{\mathrm{x}}}$ is applied in state space, which means that, as is, CL can only be included when the EnKF equations are formulated in state space. Replacing $\bar{\mathbf{P}}^{\mathsf{f}}$ by $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$ in equations (2.10a)–(2.10b)

Algorithm 7.1: Analysis step for the LEnSRF algorithm in the context of the GL system.

```
Input: $\mathbf{E}^{\mathsf{f}}\,[t_k]$, $\mathbf{y}\,[t_k]$
Parameters: $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{\rho}$
$\bar{\mathbf{x}} \leftarrow \mathbf{E}^{\mathsf{f}}\mathbf{1}/N_{\mathrm{e}}$
$\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{f}}\big(\mathbf{I}-\mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}$
$\mathbf{P} \leftarrow \boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big)$
$\mathbf{K} \leftarrow \mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}}+\mathbf{R}\big)^{-1}$
$\bar{\mathbf{x}}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}+\mathbf{K}\big(\mathbf{y}-\mathbf{H}\bar{\mathbf{x}}\big)$                // mean update
$\mathbf{T}_{\mathrm{x}} \leftarrow \mathbf{I}+\mathbf{P}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$
$\mathbf{X}^{\mathsf{a}} \leftarrow \left(\mathbf{T}_{\mathrm{x}}\right)^{-1/2}\mathbf{X}$                // perturbation update
$\mathbf{E}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{a}}\mathbf{1}^{\mathsf{T}}+\sqrt{N_{\mathrm{e}}-1}\mathbf{X}^{\mathsf{a}}$
Output: $\mathbf{E}^{\mathsf{a}}\,[t_k]$
```

and in equations (2.23a)–(2.23b), yields the following mean update:

$$\mathbf{K} = \big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}\big)\mathbf{H}^{\mathsf{T}}\Big[\mathbf{H}\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}\big)\mathbf{H}^{\mathsf{T}}+\mathbf{R}\Big]^{-1}, \tag{7.1a}$$

$$\bar{\mathbf{x}}^{\mathsf{a}} = \bar{\mathbf{x}}^{\mathsf{f}}+\mathbf{K}\big(\mathbf{y}-\mathbf{H}\bar{\mathbf{x}}^{\mathsf{f}}\big). \tag{7.1b}$$

and the following perturbation update:

$$\mathbf{T}_{\mathrm{x}} = \mathbf{I}+\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}\big)\,\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}, \tag{7.2a}$$

$$\mathbf{X}^{\mathsf{a}} = \left(\mathbf{T}_{\mathrm{x}}\right)^{-1/2}\mathbf{X}^{\mathsf{f}}. \tag{7.2b}$$

As explained in paragraph 2.2.6.3 for the global case, the matrix $\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}\big)\,\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}$ may not be symmetric, but it is diagonalisable with non-negative eigenvalues, which means that the square root of the transformation matrix $\mathbf{T}_{\mathrm{x}}$ exists and that equation (7.2b) is correctly defined. In this thesis, these update equations define the local ensemble square root Kalman filter (LEnSRF) algorithm. This is summarised in algorithm 7.1.

7.1.2 Approximate implementations

The LEnSRF algorithm is only interesting from a theoretical point of view, since it is impossible to compute the $N_{\mathrm{x}}\times N_{\mathrm{x}}$ matrix $\mathbf{T}_{\mathrm{x}}$ in high dimensional DA systems. However, there are several deterministic EnKF algorithms in which CL can be readily included.

In the deterministic ensemble Kalman filter (DEnKF) algorithm of Sakov and Oke (2008a), the perturbation update is implemented as

$$\mathbf{K} = \bar{\mathbf{P}}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\bar{\mathbf{P}}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}, \tag{7.3a}$$

$$\mathbf{X}^{\mathsf{a}} = \left[\mathbf{I} - \frac{1}{2}\mathbf{K}\mathbf{H}\right]\mathbf{X}^{\mathsf{f}}. \tag{7.3b}$$

With this perturbation update, the analysis sample covariance matrix is related to the forecast sample covariance matrix by

$$\bar{\mathbf{P}}^{\mathsf{a}} = (\mathbf{I} - \mathbf{K}\mathbf{H})\,\bar{\mathbf{P}}^{\mathsf{f}} + \frac{1}{4}\mathbf{K}\mathbf{H}\bar{\mathbf{P}}^{\mathsf{f}}\mathbf{H}^{\mathsf{T}}\mathbf{K}^{\mathsf{T}}. \tag{7.4}$$

Compared to the consistency relationship, equation (2.10c), there is an additional term. This additional term is in general non-zero, but it is positive semi-definite. This means that, even though the analysis step of the DEnKF algorithm is inconsistent, the approximation is *safe*: the analysis errors are not underestimated. The major advantage of the DEnKF algorithm is that we do not need to compute a matrix square root. From a computational point of view, this makes the algorithm very competitive. Furthermore, CL can straightforwardly be included by using $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ in place of $\bar{\mathbf{P}}^{\mathsf{f}}$.

In the serial ensemble square root filter (Whitaker and Hamill 2002), the perturbation update is based on a modified scalar Kalman gain, for which the localisation matrix $\boldsymbol{\rho}$ can be applied element-wise. Serial algorithms, however, come with their own issues, and it is also desirable to have a competitive perturbation update in matrix form.

7.1.3 Using an augmented ensemble to implement the LEnSRF

7.1.3.1 The augmented ensemble

Both the local DEnKF algorithm and the serial LEnSRF algorithm can be seen as approximate implementations of the LEnSRF algorithm. Another approximate implementation can be obtained by enlarging the ensemble size N_{e} during the analysis step. The resulting analysis step would be divided into three sub-steps.

1. Compute an ensemble $\widehat{\mathbf{E}}$, with size $\widehat{N}_{\mathrm{e}} \geq N_{\mathrm{e}}$, whose sample covariance matrix $\widehat{\mathbf{P}}$ approximates the localised sample covariance matrix $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$.

2. Use the ensemble $\widehat{\mathbf{E}}$ to compute the analysis estimate $\bar{\mathbf{x}}^{\mathsf{a}}$.

3. Use the ensemble $\widehat{\mathbf{E}}$ to compute the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$.

The ensemble $\widehat{\mathbf{E}}$ is called the **augmented ensemble**, and its size $\widehat{N}_{\mathrm{e}}$ is the called the augmented ensemble size. Augmented ensembles are already used in operational centres to implement localisation in four-dimensional ensemble variational methods (Desroziers et al. 2014, 2016; Arbogast et al. 2017).

7.1.3.2 Mean update with an augmented ensemble

Suppose that the augmented ensemble $\widehat{\mathbf{E}}$ has been computed. In this framework, the mean update is given by

$$\mathbf{K} = \widehat{\mathbf{P}}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\widehat{\mathbf{P}}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}, \tag{7.5a}$$

$$\bar{\mathbf{x}}^{\mathsf{a}} = \bar{\mathbf{x}}^{\mathsf{f}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^{\mathsf{f}}\big), \tag{7.5b}$$

where $\widehat{\mathbf{P}}$ is the sample covariance matrix of $\widehat{\mathbf{E}}$. This can be efficiently implemented, for example, using the mean update of the ETKF algorithm, algorithm 2.3.

7.1.3.3 Perturbation update with an augmented ensemble

By contrast, the perturbation update is non-trivial, because the augmented ensemble size $\widehat{N}_{\mathrm{e}}$ can be larger than the number of columns of the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$, N_{e}. However, $\widehat{\mathbf{E}}$ is constructed in such a way that $\widehat{\mathbf{P}}$ approximates $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$, in other words:

$$\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}} \approx \widehat{\mathbf{P}} = \widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}}, \tag{7.6}$$

where $\widehat{\mathbf{X}}$ is the perturbation matrix of $\widehat{\mathbf{E}}$. This can be used to compute an approximation of the transformation matrix $\mathbf{T}_{\mathrm{x}}$ as

$$\mathbf{T}_{\mathrm{x}} \approx \widehat{\mathbf{T}}_{\mathrm{x}} \triangleq \mathbf{I} + \widehat{\mathbf{X}}\widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}, \tag{7.7}$$

where $\widehat{\mathbf{Y}} = \mathbf{H}\widehat{\mathbf{X}}$. Eventually, the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ would be obtained as

$$\mathbf{X}^{\mathsf{a}} = \big(\widehat{\mathbf{T}}_{\mathrm{x}}\big)^{-1/2}\mathbf{X}^{\mathsf{f}}. \tag{7.8}$$

This perturbation update still seems intractable for high-dimensional DA systems because the transformation matrix $\widehat{\mathbf{T}}_{\mathrm{x}}$ has size $N_{\mathrm{x}} \times N_{\mathrm{x}}$. However, it can be simplified using the following theorem.

Theorem 7.1. *Let* $\mathbf{A} \in \mathbb{R}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{R}^{n\times m}$ *be two matrices such that the eigenvalues of* $\mathbf{AB}$ *and* $\mathbf{BA}$ *have a non-negative real part. Then we have the identity*

$$(\mathbf{I} + \mathbf{AB})^{-1/2} = \mathbf{I} - \mathbf{A}\Big[\mathbf{I} + \mathbf{BA} + [\mathbf{I} + \mathbf{BA}]^{1/2}\Big]^{-1}\mathbf{B}. \tag{7.9}$$

Proof. For any complex number $z \in \mathbb{C}$ which is an eigenvalue of neither $\mathbf{AB}$ nor $\mathbf{BA}$, we have the identity

$$(z\mathbf{I} - \mathbf{AB})^{-1} = \frac{1}{z}\Big[\mathbf{I} + \mathbf{A}(z\mathbf{I} - \mathbf{BA})^{-1}\mathbf{B}\Big], \tag{7.10}$$

which can be straightforwardly proven by showing that the product of the right-hand side with the inverse of the left-hand side is the identity matrix $\mathbf{I}$ and the product of the inverse of the left-hand side with the right-hand side is the identity matrix $\mathbf{I}$.

Let $f : \mathbb{C} \to \mathbb{C}$ be a function such that $f(0) = 1$, and which is analytic in a connected domain $\mathcal{D} \subset \mathbb{C}$ of contour $\mathcal{C}$ which encloses the eigenvalues of both $\mathbf{AB}$ and $\mathbf{BA}$. This is possible because the eigenvalues of $\mathbf{AB}$ and $\mathbf{BA}$ have a non-negative real part. We define the function $g : \mathbb{C} \to \mathbb{C}$ as

$$g : \begin{cases} \mathbb{C} & \to \mathbb{C}, \\ x & \mapsto \frac{f(x)-1}{x}. \end{cases} \tag{7.11}$$

Then we have the identity

$$\begin{aligned} f(\mathbf{AB}) &= \mathbf{I} + (f-1)(\mathbf{AB}), && (7.12)\\ &= \mathbf{I} + \frac{1}{2i\pi}\int_{\mathcal{C}} (f-1)(z)\,(z\mathbf{I}-\mathbf{AB})^{-1}\,\mathrm{d}z, && (7.13)\\ &= \mathbf{I} + \frac{1}{2i\pi}\int_{\mathcal{C}} (f-1)(z)\,\frac{1}{z}\Big[\mathbf{I}+\mathbf{A}(z\mathbf{I}-\mathbf{BA})^{-1}\mathbf{B}\Big]\,\mathrm{d}z, && (7.14)\\ &= \mathbf{I} + \mathbf{A}\left[\frac{1}{2i\pi}\int_{\mathcal{C}} g(z)\,(z\mathbf{I}-\mathbf{BA})^{-1}\,\mathrm{d}z\right]\mathbf{B}, && (7.15)\\ &= \mathbf{I} + \mathbf{A}g(\mathbf{BA})\mathbf{B}, && (7.16) \end{aligned}$$

where $i = \sqrt{-1}$. From the first to the second line, we apply Cauchy's integral formula of matrix argument.[1] From the second to the third line, equation (7.10) is used. From the third to the fourth line, we rely on the null contribution of the first term in the integral and the definition of the function g.

Finally, equation (7.9) can be deduced from equation (7.16) by using $f(x) = (1+x)^{-1/2}$. In this case, $g(x) = -(1+x+\sqrt{1+x})^{-1}$, and both functions are analytic in $\mathbb{C}$ except for a cut and a pole on $]-\infty, -1]$. □

Using equation (7.9) with $\mathbf{A} = \widehat{\mathbf{X}}$ and $\mathbf{B} = \widehat{\mathbf{Y}}^{\top}\mathbf{R}^{-1}\mathbf{H}$[2] yields

$$\left(\widehat{\mathbf{T}}_{\mathrm{x}}\right)^{-1/2} = \mathbf{I} - \widehat{\mathbf{X}}\left[\mathbf{I} + \widehat{\mathbf{Y}}^{\top}\mathbf{R}^{-1}\widehat{\mathbf{Y}} + \left[\mathbf{I} + \widehat{\mathbf{Y}}^{\top}\mathbf{R}^{-1}\widehat{\mathbf{Y}}\right]^{1/2}\right]^{-1}\widehat{\mathbf{Y}}^{\top}\mathbf{R}^{-1}\mathbf{H}, \tag{7.17}$$

in which the computations are mostly done in the augmented ensemble space, meaning that the matrix to invert has size $\widehat{N}_{\mathrm{e}} \times \widehat{N}_{\mathrm{e}}$. This update has been first discovered by Bocquet (2016), with a heuristic proof of theorem 7.1. It has been later rediscovered by Bishop et al. (2017) and the principle behind it has been named *gain form of the ensemble transform Kalman filter*. It is not difficult to show that their formula, equation (25), is actually mathematically equivalent to equation (25) of Bocquet (2016). However, their formula is prone to numerical cancellation errors as opposed to equation (7.17). Finally, the simple proof of theorem 7.1 reproduced here has been derived by Bocquet and Farchi (2019).

Furthermore, as shown by Bocquet and Farchi (2019), using equation (7.9) with $\mathbf{A} = \widehat{\mathbf{X}}\widehat{\mathbf{Y}}^{\top}$ and $\mathbf{B} = \mathbf{R}^{-1}\mathbf{H}$ yields

$$\left(\widehat{\mathbf{T}}_{\mathrm{x}}\right)^{-1/2} = \mathbf{I} - \widehat{\mathbf{X}}\widehat{\mathbf{Y}}^{\top}\left[\mathbf{I} + \mathbf{R}^{-1}\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^{\top} + \left[\mathbf{I} + \mathbf{R}^{-1}\widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^{\top}\right]^{1/2}\right]^{-1}\mathbf{R}^{-1}\mathbf{H}, \tag{7.18}$$

[1] It generalises the classical Cauchy's integral formula using the Jordan decomposition of matrices (see, *e.g.*, equation (2.7) of Kassam and Trefethen 2005).

[2] It can readily be checked that both **AB** and **BA** have a real and non-negative spectrum, using in particular corollary 7.6.2 of Horn and Johnson (2012), exactly as in paragraph 2.2.6.3.

Algorithm 7.2: Analysis step for the generic augmented ensemble LEnSRF algorithm in the context of the GL system.

```
  Input: $\mathbf{E}^{\mathsf{f}}\ [t_k]$, $\mathbf{y}\ [t_k]$
  Parameters: $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{\rho}$
1 $\bar{\mathbf{x}} \leftarrow \mathbf{E}^{\mathsf{f}}\mathbf{1}/N_{\mathrm{e}}$
2 $\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}$
3 $\widehat{\mathbf{X}} \leftarrow$ ensemble augmentation$(\boldsymbol{\rho}, \mathbf{X})$
4 $\widehat{\mathbf{Y}} \leftarrow \mathbf{H}\widehat{\mathbf{X}}$
5 $\widehat{\mathbf{T}}_{\mathrm{e}} \leftarrow \mathbf{I} + \widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}\widehat{\mathbf{Y}}$
6 $\mathbf{w} \leftarrow \widehat{\mathbf{T}}_{\mathrm{e}}^{-1}\widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}})$
7 $\bar{\mathbf{x}}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}} + \widehat{\mathbf{X}}\mathbf{w}$                                   // mean update
8 $\mathbf{X}^{\mathsf{a}} \leftarrow \mathbf{X} - \widehat{\mathbf{X}}\Big[\widehat{\mathbf{T}}_{\mathrm{e}} + (\widehat{\mathbf{T}}_{\mathrm{e}})^{1/2}\Big]^{-1}\widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{X}$        // perturbation update
9 $\mathbf{E}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{a}}\mathbf{1}^{\mathsf{T}} + \sqrt{N_{\mathrm{e}}-1}\mathbf{X}^{\mathsf{a}}$
  Output: $\mathbf{E}^{\mathsf{a}}\ [t_k]$
```

in which the computations are mostly done in the observation space, meaning that the matrix to invert has size $N_{\mathrm{y}} \times N_{\mathrm{y}}$.

Finally, both equations (7.17) and (7.18) support an approximation similar to that of the DEnKF algorithm, which yields

$$\mathbf{X}^{\mathsf{a}} = \mathbf{X}^{\mathsf{f}} - \frac{1}{2}\widehat{\mathbf{X}}\Big[\mathbf{I} + \widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}\widehat{\mathbf{Y}}\Big]^{-1}\widehat{\mathbf{Y}}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\mathbf{X}^{\mathsf{f}}, \tag{7.19}$$

in the first case, and

$$\mathbf{X}^{\mathsf{a}} = \mathbf{X}^{\mathsf{f}} - \frac{1}{2}\widehat{\mathbf{X}}\widehat{\mathbf{Y}}^{\mathsf{T}}\Big[\mathbf{R} + \widehat{\mathbf{Y}}\widehat{\mathbf{Y}}^{\mathsf{T}}\Big]^{-1}\mathbf{H}\mathbf{X}^{\mathsf{f}}, \tag{7.20}$$

in the second case. Again, equation (7.19) has already been proposed by Bocquet (2016), while equation (7.20) has been proposed by Bocquet and Farchi (2019).

7.1.3.4 The augmented ensemble LEnSRF algorithm

Algorithm 7.2 summarises the analysis step for the generic LEnSRF algorithm with an augmented ensemble. In this algorithm, the mean update is performed using the mean update of the ETKF algorithm in the augmented ensemble space, and the perturbation update is performed using equation (7.17), which could be replaced by equation (7.18) if the number of observations N_{y} is small. The only missing piece in this algorithm is the way to perform the ensemble augmentation (step 3). Several methods are presented in the following section.

7.2 Construction of the augmented ensemble

7.2.1 The mathematical problem

As stated in paragraph 7.1.3.1, the augmented ensemble $\widehat{\mathbf{E}}$ must be constructed in such a way that its sample covariance matrix $\widehat{\mathbf{P}}$ is an approximation of $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$. From a practical point of view, in algorithm 7.2 we only use the perturbation matrix $\widehat{\mathbf{X}}$ of $\widehat{\mathbf{E}}$. Therefore, the ensemble augmentation problem can be formulated as follows.

Problem 7.1 (Ensemble augmentation problem)**.** *Given the localised sample covariance matrix $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$, compute an $N_{\mathrm{x}} \times \widehat{N}_{\mathrm{e}}$ matrix $\widehat{\mathbf{X}}$ such that*

$$\widehat{\mathbf{X}}\mathbf{1} = \mathbf{0}, \tag{7.21a}$$

$$\widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}} \approx \boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}. \tag{7.21b}$$

In the following subsection, several methods are presented to solve this problem. For simplicity, the matrix $\widehat{\mathbf{X}}$ is called the augmented ensemble even though it represents the perturbation matrix of the augmented ensemble $\widehat{\mathbf{E}}$.

7.2.2 First method: the modulation method

Suppose that there is a matrix $\mathbf{W}$ with N_{m} columns such that the localisation matrix $\boldsymbol{\rho}$ is approximated by $\mathbf{W}\mathbf{W}^{\mathsf{T}}$. We then define the modulation product between $\mathbf{W}$ and $\mathbf{X}$ as the $N_{\mathrm{x}} \times N_{\mathrm{m}}N_{\mathrm{e}}$ matrix $\mathbf{W} \,\Delta\, \mathbf{X}$ whose elements are given by

$$\forall (n, i, j) \in (N_{\mathrm{x}} : 1) \times (N_{\mathrm{e}} : 1) \times (N_{\mathrm{m}} : 1), \quad [\mathbf{W} \,\Delta\, \mathbf{X}]_{n,(j-1)N_{\mathrm{e}}+i} = [\mathbf{W}]_{n,j}[\mathbf{X}]_{n,i}. \tag{7.22}$$

The modulation product is a mix between a Schur product (for the state variable index n) and a tensor product (for the ensemble indices i and j). As shown by Lorenc (2003), the modulation product satisfies the following factorisation property:

$$\big(\mathbf{W} \,\Delta\, \mathbf{X}\big)\big(\mathbf{W} \,\Delta\, \mathbf{X}\big)^{\mathsf{T}} = \big(\mathbf{W}\mathbf{W}^{\mathsf{T}}\big) \circ \big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big). \tag{7.23}$$

Moreover, it is easy to verify that $\mathbf{X}\mathbf{1} = \mathbf{0}$ implies $(\mathbf{W} \,\Delta\, \mathbf{X})\,\mathbf{1} = \mathbf{0}$. Therefore, $\widehat{\mathbf{X}} = \mathbf{W} \,\Delta\, \mathbf{X}$ is a solution to problem 7.1 with an augmented ensemble size $\widehat{N}_{\mathrm{e}}$ of $N_{\mathrm{m}}N_{\mathrm{e}}$ members. The name *modulation* has been given by Bishop et al. (2017). It stems from the fact that the N_{m} columns of $\mathbf{W}$ should be the main modes of $\boldsymbol{\rho}$.

Using equation (7.22), we conclude that the algorithmic complexity of computing $\widehat{\mathbf{X}}$ is $\mathcal{O}(N_{\mathrm{x}}\widehat{N}_{\mathrm{e}})$, where the complexity of computing $\mathbf{W}$ has been excluded. Indeed, if $\boldsymbol{\rho}$ is constant in time, then the same $\mathbf{W}$ can be used for all analysis steps and it only needs to be computed once. A fair comparison with the other methods must take into account this fact.

The only remaining question is how large must be the number of modes N_{m}. This question is largely discussed in the literature related to principal component analysis (see, *e.g.*, Peres-Neto et al. 2005). However, its answer highly depends on the spatial structure of the localisation matrix $\boldsymbol{\rho}$ itself. In the numerical experiments of sections 7.3 and 7.4, we illustrate how our performance criterion depends on the number of modes N_{m}. Yet at this point, it is not clear which degree of accuracy is needed for the factorisation of $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$.

7.2.3 Including balance in the modulation method

In this subsection, we describe a refinement of the modulation method based on a new idea. When there is variability between the state variables, it could be interesting to remove part of this variability by transferring it to $\mathbf{W}$ as follows. Let $\mathbf{\Lambda}$ be the $N_\mathrm{x} \times N_\mathrm{x}$ diagonal matrix containing the standard deviations of the ensemble:

$$\mathbf{\Lambda} = \Big[\mathrm{diag}\big(\mathbf{X}\mathbf{X}^\mathsf{T}\big)\Big]^{1/2}. \tag{7.24}$$

The localised sample covariance matrix $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$ can then be written

$$\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f} = \Big[\mathbf{\Lambda}\boldsymbol{\rho}\mathbf{\Lambda}\Big] \circ \Big[\big(\mathbf{\Lambda}^{-1}\mathbf{X}\big)\big(\mathbf{\Lambda}^{-1}\mathbf{X}\big)^\mathsf{T}\Big]. \tag{7.25}$$

Suppose now that the $N_\mathrm{x} \times N_\mathrm{m}$ matrix $\mathbf{W}$ verifies $\mathbf{W}\mathbf{W}^\mathsf{T} \approx \mathbf{\Lambda}\boldsymbol{\rho}\mathbf{\Lambda}$. If we have an $N_\mathrm{x} \times (N_\mathrm{m} + \delta N_\mathrm{m})$ matrix $\mathbf{V}$ such that $\mathbf{V}\mathbf{V}^\mathsf{T} \approx \boldsymbol{\rho}$, then the matrix $\mathbf{W}$ can be constructed as the N_m main modes of $\mathbf{\Lambda}\mathbf{V}$. Finally, for the same reasons as in the previous subsection, $\widehat{\mathbf{X}} = \mathbf{W}\,\Delta\,\big(\mathbf{\Lambda}^{-1}\mathbf{X}\big)$ is a solution to problem 7.1 with an augmented ensemble size $\widehat{N}_\mathrm{e}$ of $N_\mathrm{m}N_\mathrm{e}$ members.

In the transformed perturbation matrix $\mathbf{\Lambda}^{-1}\mathbf{X}$, all state variables have a unit variability. The variability transfer from $\mathbf{X}$ to $\mathbf{W}$ means that $\mathbf{W}$ can be deformed and adapted to the current situation in order to yield a more accurate approximation of $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$.

Using this method, the longest algorithmic step consists in obtaining $\mathbf{W}$ from the singular value decomposition (svd) of $\mathbf{\Lambda}\mathbf{V}$. Therefore, the algorithmic complexity of computing $\widehat{\mathbf{X}}$ is $\mathcal{O}\big(N_\mathrm{x}(N_\mathrm{m} + \delta N_\mathrm{m})^2\big)$, where, again, the cost of computing $\mathbf{V}$ has been excluded because it only needs to be computed once.

7.2.4 Second method: the truncated svd method

In this subsection, we propose an alternative to the modulation method. This new method is based on a truncated svd of $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$. We first explain how the truncated svd can be used to compute the augmented ensemble $\widehat{\mathbf{X}}$, and then we explain how to efficiently compute the truncated svd.

7.2.4.1 Construction of the augmented ensemble

Suppose that we have a truncated svd with N_m modes of $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$, which is written

$$\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f} \approx \mathbf{U}\mathbf{S}\mathbf{U}^\mathsf{T}, \tag{7.26}$$

where $\mathbf{U}$ is an $N_\mathrm{x} \times N_\mathrm{m}$ orthogonal matrix and $\mathbf{S}$ is an $N_\mathrm{m} \times N_\mathrm{m}$ diagonal matrix. Since $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$ is symmetric and positive semi-definite, equation (7.26) is a truncated eigen-decomposition as well.

Set $\varepsilon \in \{-1, 1\}$ and define $\lambda = \sqrt{N_\mathrm{m}+1}\big(\sqrt{N_\mathrm{m}+1} - \varepsilon\big)^{-1}$ and $\mathbf{Q}_\varepsilon$ as the $(N_\mathrm{m} + 1) \times$

$(N_{\mathrm{m}}+1)$ matrix whose i-th row, j-th column element is given by

$$[\mathbf{Q}_\varepsilon]_{i,j} = \begin{cases} \varepsilon/\sqrt{N_{\mathrm{m}}+1} & \text{if } i=1 \text{ or } j=1, \\ 1-\lambda/(N_{\mathrm{m}}+1) & \text{if } i=j\geq 2, \\ -\lambda/(N_{\mathrm{m}}+1) & \text{else.} \end{cases} \tag{7.27}$$

It can be easily checked that $\mathbf{Q}_\varepsilon$ is an orthogonal matrix and that $\mathbf{Q}_\varepsilon\mathbf{1}=\mathbf{e}_1$, the first basis vector. Let $\mathbf{W}$ be the $N_{\mathrm{x}}\times(N_{\mathrm{m}}+1)$ matrix whose first column is null and whose other columns are the columns of $\mathbf{US}^{1/2}$, *i.e.*, $\mathbf{W}=\big[\mathbf{0}\,,\,\mathbf{US}^{1/2}\big]$. Finally, let $\widehat{\mathbf{X}}=\mathbf{WQ}_\varepsilon$. By construction, we have

$$\widehat{\mathbf{X}}\mathbf{1}=\mathbf{0}, \tag{7.28}$$

$$\widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}}=\mathbf{WW}^{\mathsf{T}}=\mathbf{USU}^{\mathsf{T}}\approx\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}, \tag{7.29}$$

which means that $\widehat{\mathbf{X}}$ is a solution to problem 7.1 with an augmented ensemble size $\widehat{N}_{\mathrm{e}}$ of $N_{\mathrm{m}}+1$ members.

7.2.4.2 Construction of the truncated svd

The algorithmic complexity of computing the svd of the $N_{\mathrm{x}}\times N_{\mathrm{x}}$ matrix $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$ is $\mathcal{O}\big(N_{\mathrm{x}}^3\big)$. The truncated svd is then obtained by keeping the first N_{m} modes in the full svd. A more appropriate solution is probably to use the random svd algorithm derived by Halko et al. (2011).

The random svd algorithm, designed as parallelisable alternative to Lanczos techniques, is based on two ideas. In order to compute an approximate truncated svd with p columns of the matrix $\mathbf{M}\in\mathbb{R}^{m\times n}$, suppose, first, that the matrix $\mathbf{Q}$ with p orthonormal columns approximates the range of $\mathbf{M}$, *i.e.*, $\mathbf{M}\approx\mathbf{QQ}^{\mathsf{T}}\mathbf{M}$. Then an approximate truncated svd can be obtained for $\mathbf{M}$ by using the svd of the smaller matrix $\mathbf{Q}^{\mathsf{T}}\mathbf{M}$. Second, the matrix $\mathbf{Q}$ can be constructed using random draws. Indeed, if $\big\{\mathbf{x}_i, i\in(p\,{:}\,1)\big\}$ is a set of random vectors, then it is most likely a linearly independent set. Therefore, the set $\big\{\mathbf{Mx}_i, i\in(p\,{:}\,1)\big\}$ is most likely linearly independent, which means that it spans the range of $\mathbf{M}$.

One major contribution of Halko et al. (2011) and the references therein is that they have provided a mathematical justification of these ideas. In particular, they have given statistical performance bounds for their random svd algorithm and emphasised the fact that, on average, the (spectral or Frobenius) error of the resulting truncated svd should be close to the minimal error for a truncated svd with a given number of columns.

Finally, Halko et al. (2011) have introduced two elements to improve the numerical stability and efficiency of their random svd algorithm. The first element is a loop over $i\in(q\,{:}\,1)$, which forces the algorithm to construct singular vectors of $\big(\mathbf{MM}^{\mathsf{T}}\big)^q\mathbf{M}$ instead of $\mathbf{M}$. Both matrices share the same singular vectors, but the singular values of $\big(\mathbf{MM}^{\mathsf{T}}\big)^q\mathbf{M}$ decay faster than those of $\mathbf{M}$, which means that this technique enables a better approximation of the decomposition, as shown by corollary 10.10 of Halko et al. (2011). The second element is to include QR factorisations to make the algorithm less vulnerable to round-off errors. Algorithm 7.3 describes the resulting random svd algorithm, in which both elements have been taken into account. It is worth noting that algorithm 7.3 can be implemented with only

Algorithm 7.3: Random svd algorithm.

```
Input: $\mathbf{M} \in \mathbb{R}^{m \times n}$
Parameters: $p$, $q$
1  $\mathbf{Z} \overset{\text{iid}}{\sim} \mathcal{N}[\mathbf{0}, \mathbf{I}]$                              // $\mathbf{Z} \in \mathbb{R}^{n \times p}$
2  $\mathbf{B}_0 \leftarrow \mathbf{MZ}$                                         // $\mathbf{B}_0 \in \mathbb{R}^{m \times p}$
3  $\mathbf{Q}_0\mathbf{R}_0 \leftarrow \mathtt{qr}(\mathbf{B}_0)$
4  for $i = 1$ to $q$ do
5      $\widehat{\mathbf{B}}_i \leftarrow \mathbf{M}^{\mathsf{T}}\mathbf{Q}_{i-1}$                           // $\widehat{\mathbf{B}}_i \in \mathbb{R}^{n \times p}$
6      $\widehat{\mathbf{Q}}_i\widehat{\mathbf{R}}_i \leftarrow \mathtt{qr}(\widehat{\mathbf{B}}_i)$
7      $\mathbf{B}_i \leftarrow \mathbf{M}\widehat{\mathbf{Q}}_i$                                 // $\mathbf{B}_i \in \mathbb{R}^{m \times p}$
8      $\mathbf{Q}_i\mathbf{R}_i \leftarrow \mathtt{qr}(\mathbf{B}_i)$
9  end
10 $\mathbf{B} \leftarrow \mathbf{Q}_q^{\mathsf{T}}\mathbf{M}$                                       // $\mathbf{B} \in \mathbb{R}^{p \times n}$
11 $\widehat{\mathbf{U}}\mathbf{S}\mathbf{V}^{\mathsf{T}} \leftarrow \mathtt{svd}(\mathbf{B})$          // $\widehat{\mathbf{U}} \in \mathbb{R}^{p \times p}$, $\mathbf{S} \in \mathbb{R}^{p \times n}$, and $\mathbf{V} \in \mathbb{R}^{n \times n}$
12 $\mathbf{U} \leftarrow \mathbf{Q}_q\widehat{\mathbf{U}}$                                       // $\mathbf{U} \in \mathbb{R}^{m \times p}$
Output: $\mathbf{U}$, $\mathbf{S}$, $\mathbf{V}$, with $\mathbf{M} \approx \mathbf{USV}^{\mathsf{T}}$
```

the map

$$\begin{cases} \mathbb{R}^n & \rightarrow \mathbb{R}^m, \\ \mathbf{v} & \mapsto \mathbf{Mv}, \end{cases} \tag{7.30}$$

and that steps 2, 5, 7, and 10 can be parallelised by applying the matrix $\mathbf{M}$ independently to each column.

For our problem, algorithm 7.3 can be applied using the input matrix $\mathbf{M} = \boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$. The matrix-vector product in equation (7.30) can be efficiently computed by using the identity

$$\forall \mathbf{v} \in \mathbb{R}^{N_{\mathrm{x}}}, \quad \left(\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}\right) \mathbf{v} = \sum_{i=1}^{N_{\mathrm{e}}} \mathbf{X}_i \circ \left[\boldsymbol{\rho}\left(\mathbf{X}_i \circ \mathbf{v}\right)\right], \tag{7.31}$$

where $\mathbf{X}_i$ is the i-th column of the perturbation matrix $\mathbf{X}$.[3] Finally, the obtained truncated svd $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}} \approx \mathbf{USV}^{\mathsf{T}}$ is further approximated by $\mathbf{USV}^{\mathsf{T}} \approx \mathbf{USU}^{\mathsf{T}}$ because $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ is symmetric and positive semi-definite. This yields equation (7.26).

[3] Which is different from the i-th ensemble member $\mathbf{x}_i$.

7.2.4.3 Complexity of the method

In the truncated svd method, the longest algorithmic step are empirically

1. applying the localised forecast sample covariance matrix $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ (steps 2, 5, 7, and 10 in the random svd algorithm);

2. computing the QR factorisations (steps 3, 6, and 8 in the random svd algorithm).

As a consequence, the algorithmic complexity of the truncated svd method is

$$\mathcal{O}\Big(2(q+1)\widehat{N}_{\mathrm{e}}T_{\mathbf{P}} + (2q+1)N_{\mathrm{x}}\widehat{N}_{\mathrm{e}}^{2}\Big),$$

where $T_{\mathbf{P}}$ is the algorithmic complexity of applying $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ to any vector, and the parameter q is the number of iterations performed in the random svd algorithm. Using equation (7.31), we conclude that

- if $\boldsymbol{\rho}$ is banded with non-zero elements on the main N_{b} diagonals, then $T_{\mathbf{P}} = \mathcal{O}(N_{\mathrm{e}}N_{\mathrm{x}}N_{\mathrm{b}})$;

- if $\boldsymbol{\rho}$ is circulant (this corresponds to an invariance by translation in physical space), then $\boldsymbol{\rho}$ is diagonal in spectral space and $T_{\mathbf{P}} = \mathcal{O}(N_{\mathrm{e}}N_{\mathrm{x}} \ln N_{\mathrm{x}})$.

Finally, using the parallelisation potential of the random svd algorithm, we conclude that the cost of applying $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ can be reduced by a factor N_{t}, the number of threads running in parallel.

Again, the only remaining question is how large must be the number of modes N_{m}. For the same reason as in subsection 7.2.2, we cannot provide a clear answer at this point. However, in the numerical experiments of sections 7.3 and 7.4 we illustrate how our performance criterion depends on the number of modes N_{m}.

Remark 24. It is assumed here that the localisation matrix $\boldsymbol{\rho}$ is either sparse or circulant. These are sine qua none conditions for the feasibility of CL in high-dimensional DA systems.

7.3 Numerical experiments in one dimension

7.3.1 Accuracy of the augmented ensemble

We first investigate how well the methods introduced in section 7.2 solve problem 7.1, in other words how accurate is the factorisation $\widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}} \approx \boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$. To do this, a simple one-dimensional model is introduced for the forecast sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$.

7.3.1.1 A simple one-dimensional model

For any localisation radius $\ell \in \mathbb{R}_{+}$, we define $\mathbf{C}(\ell)$ as the $N_{\mathrm{x}} \times N_{\mathrm{x}}$ matrix given by

$$\forall (m,n) \in (N_{\mathrm{x}} : 1)^{2}, \quad \big[\mathbf{C}(\ell)\big]_{m,n} = G\bigg(\frac{d_{m,n}}{\ell}\bigg), \tag{7.32}$$

Table 7.1: Configuration used to construct the reference covariance matrices $\mathbf{P}^1_{\text{ref}}$ and $\mathbf{P}^2_{\text{ref}}$. The values for ℓ_c and ℓ_ref are given in number of grid points.

Parameter	Value for $\mathbf{P}^1_{\text{ref}}$	Value for $\mathbf{P}^2_{\text{ref}}$
N_x	400	400
N_e	10	10
α_c	0.25	0.25
ℓ_c	30	30
ℓ_ref	20	100

where G is the GC function introduced in subsection 2.5.3, and $d_{m,n}$ is the Euclidean distance between m and n in $(N_\text{x}\,{:}\,1)$ with periodic boundary conditions. Moreover, for any vector $\mathbf{v} \in \mathbb{R}^{N_\text{x}}$, we define $\mathbf{D}(\mathbf{v})$ as the $N_\text{x} \times N_\text{x}$ diagonal matrix whose diagonal is precisely $\mathbf{v}$.

The reference covariance matrix $\mathbf{C}_\text{ref}$ is constructed as the matrix whose correlation structure is $\mathbf{C}(\ell_\text{ref})$ and whose standard deviation vector $\mathbf{r}$ is a random draw from the distribution $\mathcal{LN}\big[\mathbf{0}, \alpha_\text{c}\,\mathbf{C}(\ell_\text{c})\big]$, where ℓ_ref, α_c, and ℓ_c are three parameters to be determined. In other words, we have

$$\mathbf{C}_\text{ref} \triangleq \mathbf{D}(\mathbf{r})\,\mathbf{C}(\ell_\text{ref})\,\mathbf{D}(\mathbf{r}). \tag{7.33}$$

We now draw a sample of N_e independent members from the distribution $\mathcal{N}[\mathbf{0}, \mathbf{C}_\text{ref}]$, and let $\mathbf{X}$ be the associated perturbation matrix. The reference (localised sample) covariance matrix is then defined as $\mathbf{P}_\text{ref} \triangleq \boldsymbol{\rho} \circ \big(\mathbf{X}\mathbf{X}^\top\big)$, with the localisation matrix $\boldsymbol{\rho} = \mathbf{C}(\ell_\text{ref})$.

In these experiments, two different reference covariance matrices $\mathbf{P}^1_{\text{ref}}$ and $\mathbf{P}^2_{\text{ref}}$ are used. They are constructed as two realisations of the model described above using the configuration described in table 7.1. The only difference between both configurations is that short-range correlations ($\ell_\text{ref} = 20$) are used to construct $\mathbf{P}^1_{\text{ref}}$ while mid-range correlations ($\ell_\text{ref} = 100$) are used to construct $\mathbf{P}^2_{\text{ref}}$. Both matrices are displayed in figure 7.1.

The modulation method described in subsections 7.2.2 and 7.2.3 requires an approximate factorisation of the localisation matrix $\boldsymbol{\rho}$, which we precompute by keeping the first N_m or $N_\text{m} + \delta N_\text{m}$ (when using the balance refinement) modes in its svd. The localisation matrix $\boldsymbol{\rho}$ being sparse, we use the random svd algorithm to obtain this factorisation.

7.3.1.2 Results and discussion

The methods described in section 7.2 are applied to obtain an approximate factorisation $\widehat{\mathbf{X}}\widehat{\mathbf{X}}^\top$ for $\mathbf{P}^1_{\text{ref}}$ and $\mathbf{P}^2_{\text{ref}}$. The accuracy of the approximate factorisation is measured with the normalised Frobenius error defined as

$$e^i_\text{F} \triangleq \frac{\left\|\mathbf{P}^i_{\text{ref}} - \widehat{\mathbf{X}}\widehat{\mathbf{X}}^\top\right\|_\text{F}}{\left\|\mathbf{P}^i_{\text{ref}}\right\|_\text{F}}, \quad i \in \{1, 2\}. \tag{7.34}$$

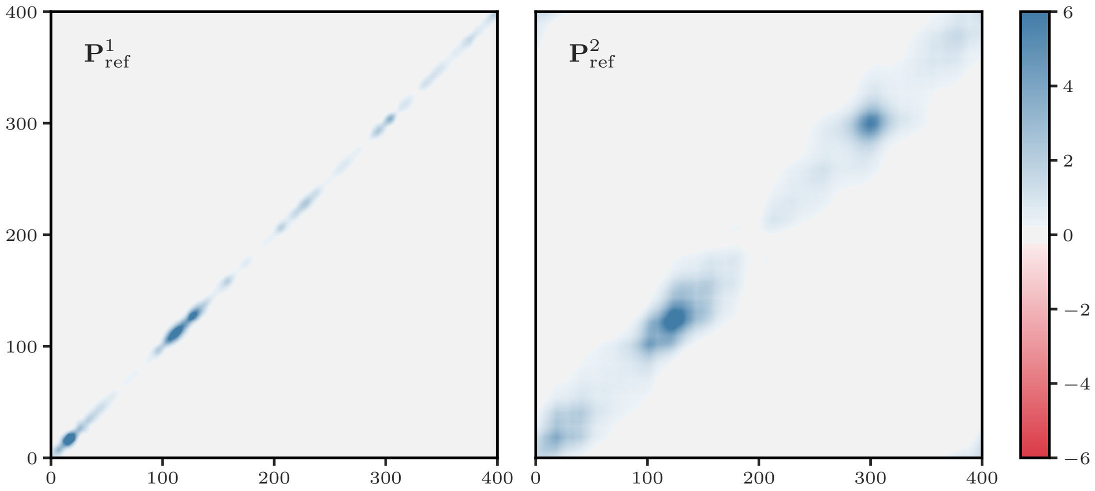

Figure 7.1: Reference covariance matrices $\mathbf{P}^1_{\text{ref}}$ (left panel) and $\mathbf{P}^2_{\text{ref}}$ (right panel).

Using the Eckart–Young theorem (Eckart and Young 1936), we conclude that the lowest normalised Frobenius error for a factorisation with rank $\widehat{N}_{\text{e}} - 1$ is

$$\left(e^i_{\text{F,min}}\right)^2 = \frac{\sum\limits_{k=\widehat{N}_{\text{e}}}^{N_{\text{x}}} \sigma_k^2\left(\mathbf{P}^i_{\text{ref}}\right)}{\left\|\mathbf{P}^i_{\text{ref}}\right\|^2_{\text{F}}}, \quad i \in \{1, 2\}, \tag{7.35}$$

where $\sigma_k(\mathbf{M})$ is the k-th singular value of the matrix $\mathbf{M}$.

Figure 7.2 shows the evolution of the normalised Frobenius error $e^{2:1}_{\text{F}}$ as a function of the augmented ensemble size $\widehat{N}_{\text{e}}$ when the factorisation is computed using the truncated svd method (subsection 7.2.4) or the modulation method with (subsection 7.2.3) or without (subsection 7.2.2) the balance refinement. The reported value is the average value over 100 independent realisations in the random svd algorithm. For $q \geq 1$ in the random svd algorithm, the Frobenius error for the truncated svd method (not illustrated here) cannot be distinguished from the minimum possible value. For the modulation method, using the balance refinement with $\delta N_{\text{m}} > 10$ (not illustrated here) does not yield a clear improvement over the case $\delta N_{\text{m}} = 10$. The singular values of $\mathbf{P}^2_{\text{ref}}$ (mid-range case) decay much faster than those of $\mathbf{P}^1_{\text{ref}}$ (short-range case). This explains why the e^2_{F} are systematically lower than the e^1_{F}.

The modulation method is very fast but yields a poor approximation of $\mathbf{P}_{\text{ref}}$. With the balance refinement, the approximation is a bit better and the method is still very fast. By contrast, the truncated svd method is much slower but yields an approximation of $\mathbf{P}_{\text{ref}}$ close to optimal. At this point, it is not clear what level of precision is needed for $\mathbf{P}_{\text{ref}}$. Yet, we can already conclude that, in a cycled DA problem, we will have to find a balance between speed and accuracy in the construction of the augmented ensemble $\widehat{\mathbf{X}}$ and in the perturbation

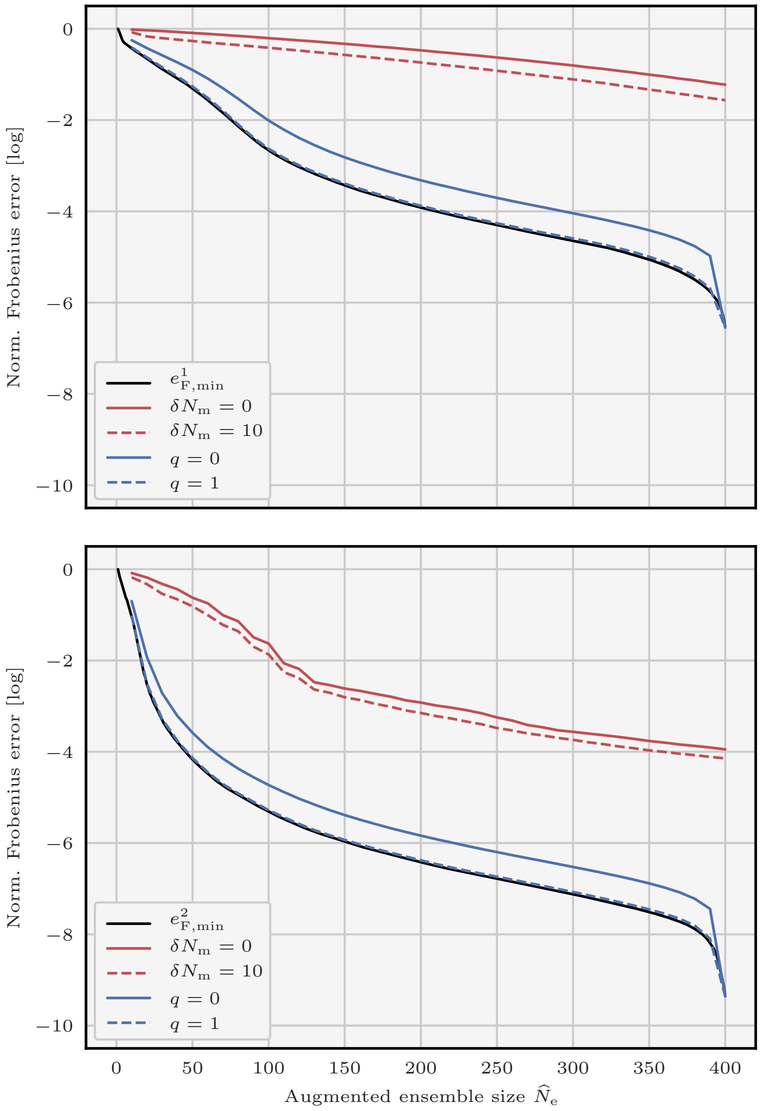

Figure 7.2: Evolution of the normalised Frobenius error e^1_{F} (top panel) and e^2_{F} (bottom panel) as a function of the augmented ensemble size $\widehat{N}_{\mathrm{e}}$. The approximate factorisation of the reference forecast sample covariance matrices $\mathbf{P}^1_{\mathrm{ref}}$ and $\mathbf{P}^2_{\mathrm{ref}}$ is computed either using the truncated svd method (in blue) for several values of the parameter q in the random svd algorithm, or using the modulation method (in red) with (continuous lines) or without (dashed lines) the balance refinement. For reference, the lowest normalised Frobenius errors $e^{2:1}_{\mathrm{F,min}}$ are plotted in black for both cases.

update.

Finally, different matrix norms could have been used in this test series. Indeed, even if equivalent, two matrix norms give different informations. This is why we have computed the normalised spectral error (not illustrated here) and found quite similar results to those depicted in figure 7.2. This shows that our results are not specific to the Frobenius norm.

7.3.2 Accuracy and efficiency of the augmented ensemble LEnSRF

In this subsection, the performance of the augmented ensemble LEnSRF algorithm, algorithm 7.2, is illustrated using twin experiments of the L96 model described in subsection 5.1.2. For these experiments, we only consider the mildly nonlinear configuration of the L96 model presented in paragraph 5.1.2.2.

7.3.2.1 Implementation notes

In this configuration, typical local domains (corresponding to typical localisation radii ℓ around 20 grid points) include all $N_\mathrm{x} = 40$ state variables. As a consequence, the localisation matrix $\boldsymbol{\rho}$ is not be sparse, which spoils the entire purpose of using the random svd algorithm. Therefore, we use $N_\mathrm{x} = 400$ state variables instead 40. The resulting configuration is essentially a repetition of ten times the original configuration. In this case, the number of unstable and neutral modes of the model dynamics is around 133.

The performance criterion is the RMSE score described in subsection 5.1.1. In order to ensure the convergence of the statistical indicators, we use a spin-up period of $N_\mathrm{s} = 2 \times 10^3$ assimilation cycles and a total simulation period of at least $N_\mathrm{c} = N_\mathrm{s} + 2 \times 10^4$ assimilation cycles.

The augmented ensemble is computed using either the truncated svd method or the modulation method with or without the balance refinement. In all cases, the ensemble size N_e is set to 10 members, less than the number of unstable and neutral modes of the model dynamics, which means that localisation is mandatory to avoid the divergence of the algorithms. The localisation matrix $\boldsymbol{\rho}$ is constructed as $\mathbf{C}(\ell)$, where ℓ is the localisation radius. This is equivalent to using equation (2.54) from subsection 2.5.3.

As presented in subsection 2.5.2, in order to mitigate the sampling errors, multiplicative inflation is used after the analysis step with a fixed multiplicative inflation factor λ. For each value of the augmented ensemble size $\widehat{N}_\mathrm{e}$, the multiplicative inflation factor λ, as well as the localisation radius ℓ, are optimally tuned to yield the lowest RMSE score.

For the modulation method, the approximate factorisation of $\boldsymbol{\rho}$ is precomputed once for each twin experiment by keeping the first N_m of $N_\mathrm{m} + \delta N_\mathrm{m}$ (when using the balance refinement) modes in the svd of $\boldsymbol{\rho}$. Finally for the truncated svd method, the matrix multiplication with $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathrm{f}$ are computed using equation 7.31, and $\boldsymbol{\rho}$ is applied in spectral space.

7.3.2.2 Accuracy of the algorithm

Figure 7.3 shows the evolution of the RMSE score as a function of the augmented ensemble size $\widehat{N}_\mathrm{e}$. Both the truncated svd and the modulation methods are able to produce an estimate of the truth $\mathbf{x}^\mathrm{t}$ with an accuracy equivalent to that of the LETKF algorithm, algorithm 2.4. As expected after the experiments of subsection 7.3.1, for a given level of RMSE score we need

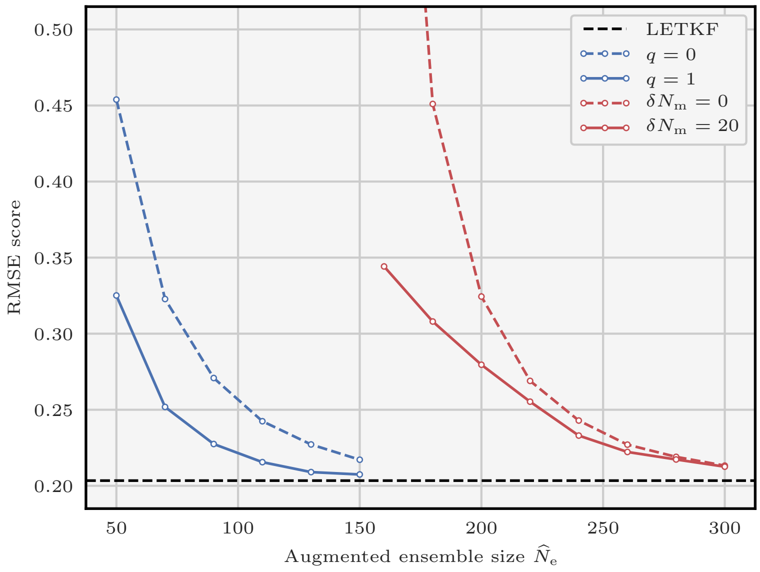

Figure 7.3: Evolution of the RMSE score as a function of the augmented ensemble size $\widehat{N}_{\mathrm{e}}$ for the augmented ensemble LEnSRF algorithm. The augmented ensemble $\widehat{\mathbf{X}}$ is computed either using the truncated svd method (in blue) for several values of the parameter q in the random svd algorithm, or using the modulation method (in red) with (continuous lines) or without (dashed lines) the balance refinement. For reference, the RMSE score of the LETKF algorithm with an ensemble of $N_{\mathrm{e}} = 10$ members is shown with an horizontal dashed black line. The DA system is the L96 model in the extended mildly nonlinear configuration.

a much smaller $\widehat{N}_{\mathrm{e}}$ when using the truncated svd method than when using the modulation method. However, before we conclude that the truncated svd method is more efficient, we need to take into account the fact that computing the augmented ensemble $\widehat{\mathbf{X}}$ is much slower with this method than with the modulation method.

With the truncated svd method, the augmented ensemble size $\widehat{N}_{\mathrm{e}}$ needs to be at least of the same order as the number of unstable and neutral modes of the model dynamics in order to yield accurate results. This coincides with the results of Bocquet et al. (2017). We have also tested $q > 1$ in the truncated svd method and $\delta N_{\mathrm{m}} > 20$ in the modulation method (not illustrated here) and found that none of these parameters yields clear improvements in RMSE scores.

7.3.2.3 Efficiency of the algorithm

In the augmented ensemble LEnSRF algorithm, the longest algorithmic steps are empirically

- the construction of the augmented ensemble $\widehat{\mathbf{X}}$;
- the inverse and the inverse square root of the transformation matrix $\widehat{\mathbf{T}}_{\mathrm{e}}$.

Regarding the second point, we have tested several approaches and concluded that the

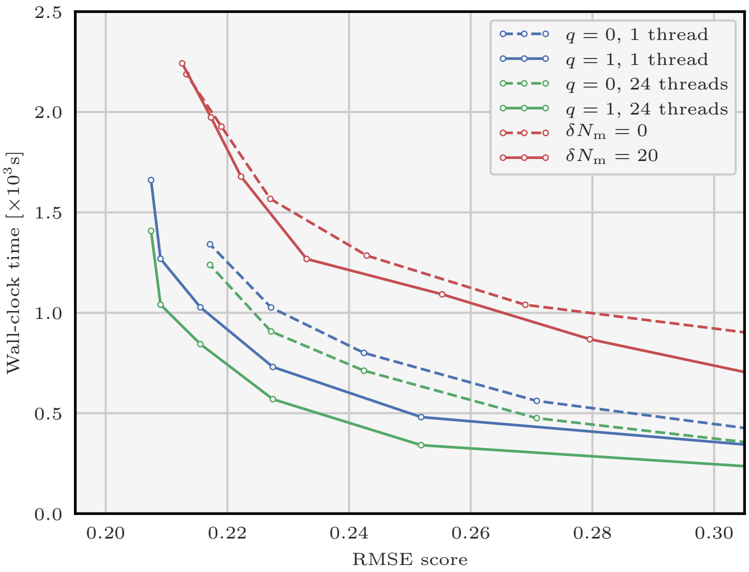

Figure 7.4: Evolution of the wall-clock analysis time for the 22×10^3 assimilation cycles as a function of the RMSE score for the augmented ensemble LEnSRF algorithm. The augmented ensemble $\widehat{\mathbf{X}}$ is computed either using the truncated svd method (in blue and in green) for several values of the parameter q in the random svd algorithm, or using the modulation method (in red) with (continuous lines) or without (dashed lines) the balance refinement. The DA system is the L96 model in the extended mildly nonlinear configuration.

most efficient is to compute the svd of $\mathbf{R}^{-1/2}\widehat{\mathbf{Y}}$ using a direct svd algorithm, which cannot be parallelised. Regarding the first point, some level of parallelisation can be included in the random svd algorithm (*i.e.*, when using the truncated svd method). When using the modulation method (even with the balance refinement), the construction of $\widehat{\mathbf{X}}$ is almost instantaneous compared to the svd of $\mathbf{R}^{-1/2}\widehat{\mathbf{Y}}$. Therefore, we only enable parallelisation in the random svd algorithm.

Figure 7.4 shows the evolution of the wall-clock time of one analysis step as a function of the RMSE score. All experiments are performed on the same computational platform with 12 cores. Parallelisation is enabled when possible using a fixed number of `OpenMP` threads. For a given level of RMSE score, the truncated svd method is clearly faster than the modulation method. This shows the advantage of using the truncated svd method over the modulation method, especially when parallelisation is possible. However, this result is specific to the problem considered here and may not generalise to all situations.

7.4 Numerical experiments with satellite radiances

7.4.1 Is covariance localisation viable in high-dimensional DA systems?

In subsection 7.3.2, we have implemented CL in the EnKF and successfully applied the resulting algorithm to a one-dimensional DA system with $N_{\mathrm{x}} = 400$ state variables. With a high-dimensional system, CL in the EnKF will probably require a very large augmented ensemble size $\widehat{N}_{\mathrm{e}}$, too large to be affordable. In this case, the use of DL will be mandatory.

When observations are local, DL is simple to implement and yield efficient algorithms such as the LETKF algorithm. However, when observations are non local, one must resort to *ad hoc* approximations to implement DL in the EnKF, for example assigning an approximate location to each observation. In this section, we discuss the case of satellite radiances, which are non-local observations, and we show how existing variants of the LETKF algorithm deal with such observations. We then give an generalisation of the augmented ensemble LEnSRF algorithm designed to assimilate satellite radiances in a spatially extended model. Finally we introduce a multilayer extension of the L96 model which mimics satellite radiances and we illustrate the performance of the generalised augmented ensemble LEnSRF algorithm using twin experiments of this multilayer model.

7.4.2 The case of satellite radiances

Suppose that the physical space consists of a multilayer space with P_{z} vertical levels of P_{h} state variables. For any $h \in (P_{\mathrm{h}}\!:\!1)$ and $z \in (P_{\mathrm{z}}\!:\!1)$, the state variable located at the h-th (horizontal) grid point and at the z-th vertical level is written $x_{(z,h)}$. For any vector $\mathbf{x}$, the sub-vector containing the P_{z} elements of $\mathbf{x}$ which are located at the h-th grid point is written $\mathbf{x}_h$ and called the h-th **column** of $\mathbf{x}$. Suppose furthermore that each column of the state vector $\mathbf{x}$ is independently observed through

$$\mathbf{y}_h = \boldsymbol{\Omega}\mathbf{x}_h, \tag{7.36}$$

where $\boldsymbol{\Omega}$ is a $P_{\mathrm{c}} \times P_{\mathrm{z}}$ weighting matrix, $\mathbf{y}_h$ is the vector containing the P_{c} observations at the h-th grid point, and P_{c} is the number of **channels**. The full observation vector $\mathbf{y}$ is the concatenation of all $\mathbf{y}_{P_{\mathrm{h}}:1}$. It has $N_{\mathrm{y}} = P_{\mathrm{c}} \times P_{\mathrm{h}}$ elements and for any $h \in (P_{\mathrm{h}}\!:\!1)$ and $c \in (P_{\mathrm{c}}\!:\!1)$, the observation located at the h-th grid point and corresponding to the c-th channel is written $y_{(c,h)}$.

This simple model describes a typical situation for satellite radiances. From these definitions, it is clear that each observation is attached to an horizontal position, but has no well-defined vertical position (unless the weighting matrix $\boldsymbol{\Omega}$ is diagonal). Several variants of the LETKF algorithm have been designed to assimilate such observations. When the weighting function of each channel has a single and well-located maximum, the vertical location of this maximum can play the role of an approximate height for the channel. This is the approach followed for example by Fertig et al. (2007). Based on these vertical positions, they use the channels to update *adjacent* vertical levels as long as the corresponding weighting function is above a threshold value. Campbell et al. (2010) has followed the same approach to define the approximate height of the channels. However their update formula includes a vertical tapering of the anomalies depending on the vertical distance. When the weighting functions are flat, another possibility is to define the approximate height of each channel as the middle of the support of its weighting function (Anderson and Lei 2013). Miyoshi and Sato (2007) have proposed an alternative which does not require the definition of an approximate height of the channels: their update formula includes a vertical tapering of the anomalies which depends on the shape of the weighting functions only. Finally, in the algorithm of Penny et al. (2015), vertical localisation has simply been removed.

Using a realistic one-dimensional model with satellite radiances, Campbell et al. (2010) have shown that *ad hoc* approaches based on DL only systematically yield higher errors than CL. In a spatially extended system with satellite radiances, it seems natural to apply DL in the horizontal direction, in which observations are local, while using CL in the vertical direction, in which observations are non-local.

7.4.3 Including domain localisation in the LEnSRF

Following the approach of Bishop et al. (2017), we apply four modifications to the augmented ensemble LEnSRF algorithm (algorithm 7.2) in order to include DL in way similar to the LETKF algorithm.

1. We perform P_{h} local analyses instead of one global analysis. The aim of the h-th local analysis is to give an update for the P_{z} state variables which form the h-th column. The linear algebra must be amended accordingly.

2. We taper the anomalies related to each observation with respect to the *horizontal* distance to the h-th column. This is usually implemented in $\mathbf{R}^{-1/2}$.

3. Observations whose horizontal position is far from the h-th column (that is, observations whose site is not located in the h-th local domain) do not contribute to the update. These observations are therefore omitted in the local analysis in order to save some computational time.

4. Since observations located outside of the local domain are omitted, we only need to compute an augmented ensemble $\widehat{\mathbf{X}}$ for the state variables inside the local domain. Since CL is only applied in the vertical direction, the (local) augmented ensemble $\widehat{\mathbf{X}}^{\ell}$ must be constructed in such a way that

$$\widehat{\mathbf{X}}^{\ell}\big(\widehat{\mathbf{X}}^{\ell}\big)^{\mathsf{T}} \approx \boldsymbol{\rho}_{\mathrm{v}} \circ \Big[\mathbf{X}^{\ell}\big(\mathbf{X}^{\ell}\big)^{\mathsf{T}}\Big], \tag{7.37}$$

   where $\mathbf{X}^{\ell}$ is the restriction of the perturbation matrix $\mathbf{X}$ to the local domain, and $\boldsymbol{\rho}_{\mathrm{v}}$ is the vertical localisation matrix, whose elements only depend on the vertical layer indices.

The resulting algorithm, hereafter called the local analysis LEnSRF (L^{2}EnSRF) algorithm, implements DL in the horizontal direction and CL in the vertical direction. Therefore, it can be used to assimilate vertically non-local observations such as satellite radiances.

7.4.4 The multilayer L96 model

We now introduce a multilayer extension of the L96 model, hereafter called multilayer Lorenz 1996 (mL96) model. This multilayer extension is used to illustrate the performance of the L^{2}EnSRF algorithm.

The mL96 model consists of P_z coupled layers of the one-dimensional L96 model with P_h variables. Keeping the notations defined in subsection 7.4.2, the evolution of the h-th variable of the z-th level in the model is given by the following ODE:

$$\begin{aligned}\frac{\mathrm{d}x_{(z,h)}}{\mathrm{d}t} = \big[x_{(z,h+1)} - x_{(z,h-2)}\big]\, x_{(z,h-1)} - x_{(z,h)} + F_z& \\ + \delta_{\{z>0\}}\,\Gamma\big[x_{(z-1,h)} - x_{(z,h)}\big]& \\ + \delta_{\{z\leq P_\mathrm{z}\}}\,\Gamma\big[x_{(z+1,h)} - x_{(z,h)}\big]&. \qquad (7.38)\end{aligned}$$

The first line in equation (7.38) corresponds to the ODE of the original L96 model, equation (5.2), with a forcing term F which may depend on the vertical layer index z. The second and third lines correspond to the coupling between adjacent layers, with a constant intensity Γ. As for the L96 model, the horizontal indices are to be understood with periodic boundary conditions:

$$\forall z \in (P_\mathrm{z}\!:\!1), \quad x_{(z,-1)} = x_{(z,P_\mathrm{h}-1)}, \quad x_{(z,0)} = x_{(z,P_\mathrm{h})}, \quad \text{and} x_{(z,1)} = x_{(z,P_\mathrm{h}+1)}. \qquad (7.39)$$

As for the L96 model, the ODEs are integrated using a fourth-order Runge–Kutta method with an integration time step δt of 0.05 unit of time.

For these experiments, we use $P_\mathrm{z} = 32$ layers and $P_\mathrm{h} = 40$ to mimic the mildly nonlinear configuration of the L96 model. The forcing term F linearly decreases from $F_1 = 8$ at the lowest level to $F_{P_\mathrm{z}} = 4$ at the highest level. Without the coupling, these values would render the lower levels dynamics chaotic and the higher levels dynamics laminar, which is a typical behaviour in the atmosphere. Finally, we set $\Gamma = 1$ such that adjacent layers are highly correlated (correlation around 0.87). To be more specific, the correlation between the z-th level and the $(z+\delta z)$-th level first rapidly decreases with δz. It reaches approximately -0.1 for $\delta z = 6$ layers and then it starts increasing. Finally, its absolute value is below 10^{-2} when $\delta z > 10$ layers. This model is chaotic and the dimension of the unstable or neutral subspace is around 50.

The observation operator $\mathbf{H}$ follows the model described in subsection 7.4.2. We use $P_\mathrm{c} = 8$ channels and a weighting matrix $\boldsymbol{\Omega}$ designed to mimic satellite radiances, as shown in figure 7.5. The observation vectors are given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{e}^\mathrm{o}, \quad \mathbf{e}^\mathrm{o} \sim \mathcal{N}[\mathbf{0}, \mathbf{I}], \qquad (7.40)$$

and the time interval between consecutive observations Δt is the same as the one used with the L96 model, 0.05 unit of time. Again, the standard deviation of the observation noise is approximately one tenth of the climatological variability of each observation.

For the horizontal localisation, we use the Euclidean distance d_h in $(P_\mathrm{h}\!:\!1)$ with periodic boundary conditions. For the vertical localisation, we use the Euclidean distance d_v in $(P_\mathrm{z}\!:\!1)$.

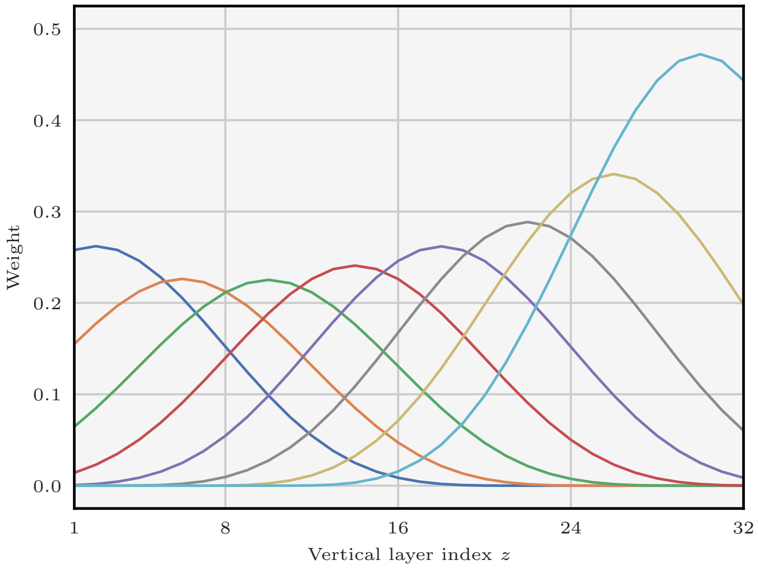

Figure 7.5: Observation operator **H** used with the mL96 model. Each line represents the weighting function of a different channel, corresponding to a row of the weighting matrix $\boldsymbol{\Omega}$. Every channel has a single maximum and is relatively broad (half-width around 10 vertical layers). The sum of the weights has been adjusted individually such that every channel yields an observation with approximately the same climatological variability.

7.4.5 Implementation notes

In this section, we give some details on how localisation is implemented in the L^{2}EnSRF algorithm for the mL96 model. We then describe the approximations necessary to implement an *ad hoc* LETKF algorithm.

7.4.5.1 Horizontal localisation

Let ℓ_h be the horizontal localisation radius. During the h_1-th local analysis, the anomalies related to the c-th channel of the observation vector at the h_2-th grid point, $y_{(c,h_2)}$ are tapered by a factor

$$\sqrt{G\bigg(\frac{2d_\mathrm{h}(h_1,h_2)}{\ell_\mathrm{h}}\bigg)},$$

where G is the GC function introduced in subsection 2.5.3. This means that the h-th local domain consists of the columns $\{h-\lfloor\ell_\mathrm{h}\rfloor,\dots,h+\lfloor\ell_\mathrm{h}\rfloor\}$ where the indices are to be understood with periodic boundary conditions in $(P_\mathrm{h}:1)$, and where $\lfloor\ell_\mathrm{h}\rfloor$ is the integer part of ℓ_h.

7.4.5.2 Vertical localisation

Let ℓ_v be the vertical localisation radius. The (z_1,h_1)-th row, (z_2,h_2)-th column element of the vertical localisation matrix $\boldsymbol{\rho}_\mathrm{v}$ is given by

$$[\boldsymbol{\rho}_\mathrm{v}]_{(z_1,h_1),(z_2,h_2)} = G\bigg(\frac{2d_\mathrm{v}(z_1,z_2)}{\ell_\mathrm{v}}\bigg). \tag{7.41}$$

The local domains gather $P^\ell_\mathrm{h}=2\lfloor\ell\rfloor+1$ columns, hence $\boldsymbol{\rho}_\mathrm{v}$ is a $P_\mathrm{z}P^\ell_\mathrm{h}\times P_\mathrm{z}P^\ell_\mathrm{h}$ block-diagonal matrix. Since its elements only depend on the vertical layer indices, it can also be seen as a $P_\mathrm{z}\times P_\mathrm{z}$ matrix.

The $P_\mathrm{z}P^\ell_\mathrm{h}\times\widehat{N}_\mathrm{e}$ matrix $\widehat{\mathbf{X}}^\ell$ of the (local) augmented ensemble is computed using either the truncated svd method or the modulation method with or without the balance refinement. In all cases, the ensemble size N_e is set to 8 members.

For the modulation method, the approximate factorisation of $\boldsymbol{\rho}_\mathrm{v}$ is precomputed once for each experiment by keeping the first N_m or $N_\mathrm{m}+\delta N_\mathrm{m}$ (when using the balance refinement) modes in the svd of the $P_\mathrm{z}\times P_\mathrm{z}$ matrix $\boldsymbol{\rho}_\mathrm{v}$. Finally for the truncated svd method, the matrix multiplications with $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^\mathsf{f}$ are computed using equation (7.31). Since the elements of $\boldsymbol{\rho}_\mathrm{v}$ only depend on the vertical layer indices, applying the $P_\mathrm{z}P^\ell_\mathrm{h}\times P_\mathrm{z}P^\ell_\mathrm{h}$ matrix $\boldsymbol{\rho}_\mathrm{v}$ to a vector with $P_\mathrm{z}P^\ell_\mathrm{h}$ elements reduces to applying the $P_\mathrm{z}\times P_\mathrm{z}$ matrix $\boldsymbol{\rho}_\mathrm{v}$ to a vector with P_z elements. It should be relatively quick and therefore we do not perform this product in spectral space. This means that the implementation can be straightforwardly generalised to the general case where the vertical layers are not equally distributed in height.

7.4.5.3 Approximations for the LETKF

We define the approximate height z_c of the c-th channel as

$$z_c \triangleq \frac{\displaystyle\sum_{z=1}^{P_\mathrm{z}} z[\mathbf{\Omega}]_{c,z}}{\displaystyle\sum_{z=1}^{P_\mathrm{z}} [\mathbf{\Omega}]_{c,z}} \in [1, P_\mathrm{z}]. \tag{7.42}$$

We did not define the c-th approximate height z_c as the vertical position of the maximum of the c-th weighting function because we wanted to account for the fact that our weighting functions are skewed in the vertical direction.

In the *ad hoc* LETKF algorithm, $P_\mathrm{z} \times P_\mathrm{h}$ local analyses are performed, one for each state variable. Furthermore, in the local analysis for the variable at the h_1-th grid point of the z-th level, $x_{(z,h_1)}$, the anomalies related to the c-th channel of the observation vector at the h_2-th grid point, $y_{(c,h_2)}$, are tapered by a factor

$$\sqrt{G\left[2\sqrt{\delta h^2/\ell_\mathrm{h}^2 + \delta z^2/\ell_\mathrm{v}^2}\right]},$$

where

$$\delta h = \min\bigl\{|h_2 - h_1|, P_\mathrm{h} - |h_2 - h_1|\bigr\}, \tag{7.43}$$
$$\delta z = |z - z_c|. \tag{7.44}$$

7.4.6 Results

For this experiment, the performance criterion is the RMSE score described in subsection 5.1.1. In order to ensure the convergence of the statistical indicators, we use a spin-up period of $N_\mathrm{s} = 10^3$ assimilation cycles and a total simulation period of at least $N_\mathrm{c} = N_\mathrm{s} + 10^4$ assimilation cycles. As presented in subsection 2.5.2, in order to mitigate the sampling errors, multiplicative inflation is used after the analysis step with a fixed multiplicative inflation factor λ.

Figure 7.6 shows the evolution of the RMSE score and of the wall-clock time of one analysis step as a function of the horizontal localisation radius ℓ_h for the L^{2}EnSRF and the LETKF algorithms. For each value of the horizontal localisation radius ℓ_h, the multiplicative inflation factor λ, as well as the vertical localisation radius ℓ_v, are optimally tuned to yield the lowest RMSE score. All experiments are performed on the same computational platform with 12 cores. Parallelisation is enabled for the P_h independent local analyses using a fixed number of `OpenMP` threads $N_\mathrm{t} = 20$. In the L^{2}EnSRF algorithm, the augmented ensemble $\widehat{\mathbf{X}}$ is computed using either the truncated svd method with $q = 0$ in the random svd algorithm, or the modulation method without the balance refinement ($\delta N_\mathrm{m} = 0$). Preliminary experiments with $q > 0$ or $\delta N_\mathrm{m} > 0$ (not illustrated here) did not display clear improvements in RMSE score over the cases $q = 0$ and $\delta N_\mathrm{m} = 0$.

The *ad hoc* LETKF algorithm yields rather high RMSE scores (compared to the observation

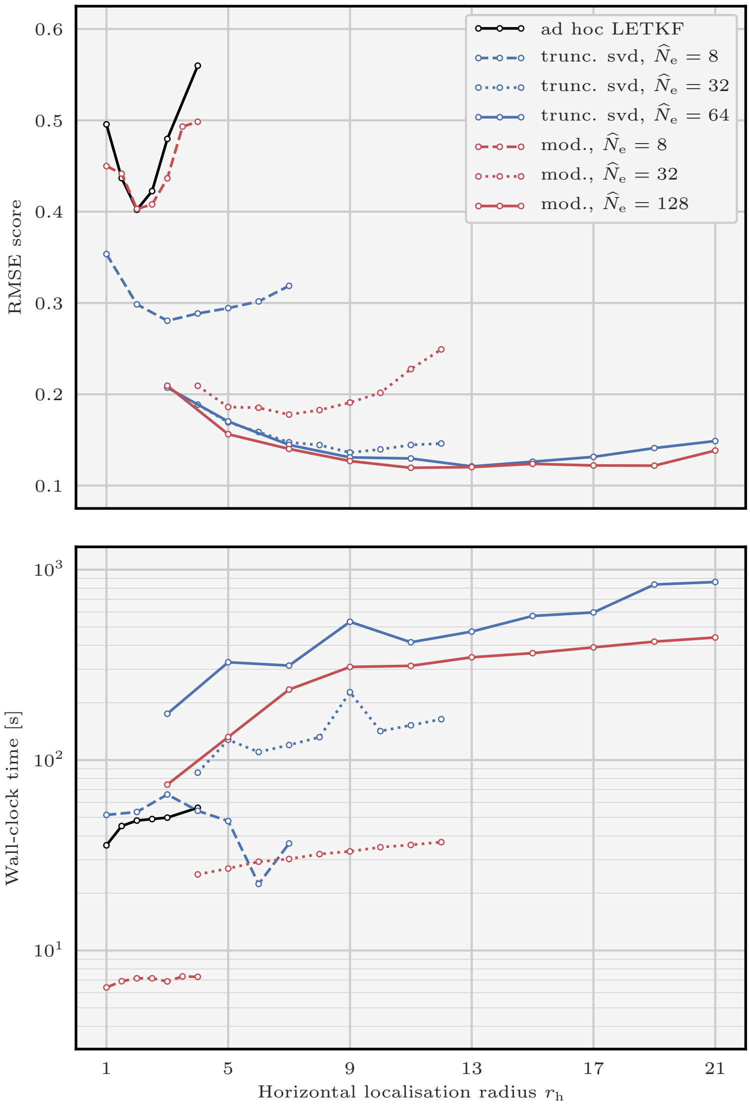

Figure 7.6: Evolution of the RMSE score (top panel) and of the wall-clock analysis time for the 10^3 cycles (bottom panel) as a function of the horizontal localisation radius ℓ_h for the L^{2}EnSRF algorithm (in blue and in red) and for the *ad hoc* LETKF algorithm (in black). The augmented ensemble is computed using either the truncated svd method (in blue) with $q = 0$ in the random svd algorithm, or the modulation method (in red) without the balance refinement ($\delta N_m = 0$). In both cases, several values of the augmented ensemble size $\widehat{N}_e$ are tested. The DA system is the mL96 model described in subsection 7.4.4.

standard deviation $r = 1$), while not completely failing to reconstruct the truth $\mathbf{x}^{\mathrm{t}}$. Although DL in the horizontal direction is a powerful tool, vertical localisation is necessary in this DA system. Because the weighting functions of the channels are quite broad, observations cannot be considered local and DL in the vertical direction is inefficient. By contrast, with a reasonable augmented ensemble size $\widehat{N}_{\mathrm{e}}$, the L^{2}EnSRF algorithm yields significantly lower RMSE scores. This shows that combining DL in the horizontal direction and CL in the vertical direction is an adequate approach to assimilate satellite radiances.

The comparison between the truncated svd and the modulation methods is not as simple as it was in the experiments with the L96 model. As expected, for a given augmented ensemble size $\widehat{N}_{\mathrm{e}}$, the truncated svd method yields lower RMSE scores. However, for a given level of RMSE score, using the truncated svd method is not always the fastest approach. For example, the RMSE score for the truncated svd method with $\widehat{N}_{\mathrm{e}} = 64$ is approximately the same as the RMSE score for the modulation method with $\widehat{N}_{\mathrm{e}} = 128$, but in this case the modulation method is faster by a factor 1.5 on average. This can be explained by two factors. First, in the truncated svd method $\boldsymbol{\rho}_{\mathrm{v}}$ is not applied in spectral space. Second, both the truncated svd and the modulation method benefit from parallelisation, but the parallelisation potential of the truncated svd method is not fully exploited here because our computational platform has a limited number of threads. This would change if we could use several threads per local analysis. Finally, these results confirm that, for high dimensional DA systems where the memory requirement is an issue, the truncated svd method is the best approach to obtain accurate results while using only a limited augmented ensemble size $\widehat{N}_{\mathrm{e}}$.

7.5 Summary and discussion

In this chapter, we have explored possible implementations for CL in deterministic EnKF algorithms using an augmented ensemble in the analysis step. We have discussed the two main difficulties related to the use of augmented ensembles: how to construct the augmented ensemble and how to update the perturbations.

We have used two different methods to construct the augmented ensemble. The first one is based on a factorisation property of the forecast sample covariance matrix. It is already widespread in the geophysical DA literature under the name *modulation*. For this method, we have also introduced a *balance* refinement in order to smooth some variability between the state variables. As an alternative, we have proposed a second method based on randomised svd techniques, which are very efficient when the localisation matrix is easy to apply. The random svd algorithm, is commonly used in the statistical literature but it had never been applied in this context. We have called this approach the *truncated svd* method.

We have shown how CL can be included in the perturbation update using the augmented ensemble framework. The resulting update formula (Bocquet 2016) uses linear algebra in the augmented ensemble space. It is included in the generic augmented ensemble LEnSRF algorithm.

Using numerical experiments with a very simple one-dimensional covariance model with 400 state variables, we have shown that the truncated svd method yields a much more accurate approximation of the (localised) forecast sample covariance matrix than the modulation method. This result has been confirmed by twin experiments using the one-dimensional L96 model with 400 variables. In an extended mildly nonlinear configuration of the L96 model, we have found that the balance between fast computation of the augmented ensemble and fast perturbation update is in favour of the truncated svd method. In other words, for a given level of RMSE score, it is worth spending more time to construct a smaller but more reliable augmented ensemble with the truncated svd method and then use a faster perturbation update.

We have defined the L^{2}EnSRF algorithm as a generalisation of the LEnSRF algorithm suited to the assimilation of satellite radiances in spatially extended models. It implements DL in the horizontal direction in a similar way as the LETKF algorithm, and CL in the vertical direction. Such an extension had been discussed by Bishop et al. (2017) but without numerical illustration.

Finally, we have constructed a simple multilayer extension of the L96 model, called the mL96 model. We have performed twin experiments with this model using a satellite-like observation operator. As expected, the LETKF algorithm hardly reconstructs the truth. By contrast, the L^{2}EnSRF algorithm yields an estimate of the truth with an acceptable accuracy. We have concluded that using DL in the horizontal direction and CL in the vertical direction is an adequate approach to assimilate satellite radiances in a spatially extended model. For a given level of RMSE score, the modulation method is the fastest approach in this DA system. This result is mitigated by the fact that our computational setup does not use the full parallelisation potential of the truncated svd method. When the augmented ensemble size is limited, the truncated svd method is the best approach to obtain accurate results.

8 Consistency of the LEnSRF perturbation update

Contents

Theorem 2.5 ensures that the analysis step of the ETKF algorithm is consistent according to the notion of consistency introduced in subsection 2.2.2. When using CL in the EnKF, the notion of consistency has to be redefined to take into account the fact that the forecast sample covariance matrix is localised.

In this chapter, following the work published in Bocquet and Farchi (2019), we revisit the perturbation update of deterministic EnKF algorithms using CL, with a focus on the consistency. In section 8.1, we introduce a notion of consistency coherent with the use of CL. The consistency of the LEnSRF perturbation update is then discussed in details, and a new approach is proposed. The algorithmic complexity of this new approach is explicitly computed. In section 8.2, the new approach is implemented and tested using twin experiments of low-order one-dimensional models. Finally, conclusions are given in section 8.3. In this chapter, unless specified otherwise, the DA system is the GL system. For simplicity, the time subscript k is systematically dropped in the equations.

8.1 A new perturbation update method

In chapter 7, we have defined the LEnSRF algorithm and explained how it could be implemented, for example by using an augmented ensemble. In this section, we focus on the perturbation update step.

8.1.1 On the consistency of the perturbation update

When using CL, the forecast sample covariance matrix $\bar{\mathbf{P}}^{\mathsf{f}}$ is replaced by its localised version $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$. The consistency relationships for the EnKF analysis step, defined in subsection 2.2.2 by equations (2.10a)–(2.10c), has to be amended accordingly. Replacing $\bar{\mathbf{P}}^{\mathsf{f}}$ by $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ in equations (2.10a)–(2.10c) yields

$$\mathbf{K} = (\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}})\mathbf{H}^{\mathsf{T}}\Big[\mathbf{H}(\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}})\mathbf{H}^{\mathsf{T}} + \mathbf{R}\Big]^{-1}, \tag{8.1a}$$

$$\bar{\mathbf{x}}^{\mathsf{a}} = \bar{\mathbf{x}}^{\mathsf{f}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}^{\mathsf{f}}\big), \tag{8.1b}$$

$$\bar{\mathbf{P}}^{\mathsf{a}} = \big(\mathbf{I} - \mathbf{K}\mathbf{H}\big)\big(\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}\big). \tag{8.1c}$$

These equations define the consistency relationships for the EnKF analysis step with CL. Furthermore, using the Sherman–Morrison–Woodbury matrix identity in equation (8.1c) yields

$$\bar{\mathbf{P}}^{\mathsf{a}} = \Big[\mathbf{I} + (\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}})\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}\Big]^{-1}\big(\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}\big). \tag{8.2}$$

In order to avoid any confusion, it is assumed in this chapter that $\bar{\mathbf{P}}^{\mathsf{a}}$ is the estimated analysis error covariance matrix defined by equation (8.2). The EnKF perturbation update is said to be consistent if the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ is related to $\bar{\mathbf{P}}^{\mathsf{a}}$ through

$$\bar{\mathbf{P}}^{\mathsf{a}} = \mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}. \tag{8.3}$$

As explained in paragraph 7.1.3.3, in the augmented ensemble framework, $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}}$ can be approximated by $\widehat{\mathbf{P}} = \widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}}$, the sample covariance matrix of the augmented ensemble $\widehat{\mathbf{X}}$. If the augmented ensemble size $\widehat{N}_{\mathrm{e}}$ is (strictly) larger than N_{x}, it is possible to construct $\widehat{\mathbf{X}}$ in such a way that the approximation is exact, that is $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^{\mathsf{f}} = \widehat{\mathbf{P}}$. In this case, the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathsf{a}}$ can be exactly factorised as

$$\bar{\mathbf{P}}^{\mathsf{a}} = \widehat{\mathbf{X}}^{\mathsf{a}}\big(\widehat{\mathbf{X}}^{\mathsf{a}}\big)^{\mathsf{T}}, \tag{8.4}$$

where the $N_{\mathrm{x}} \times \widehat{N}_{\mathrm{e}}$ analysis perturbation matrix $\widehat{\mathbf{X}}^{\mathsf{a}}$ is obtained using

$$\mathbf{T}_{\mathrm{x}} = \mathbf{I} + \widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}}\mathbf{H}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{H}, \tag{8.5a}$$

$$\widehat{\mathbf{X}}^{\mathsf{a}} = (\mathbf{T}_{\mathrm{x}})^{-1/2}\widehat{\mathbf{X}}. \tag{8.5b}$$

This perturbation update is by construction consistent because it satisfies equation (8.3).

Of course, this is only theoretical since, in practice, we can only afford to generate and propagate an ensemble of size $N_{\mathrm{e}} \ll N_{\mathrm{x}}$. Since we look for N_{e} members which capture most of the uncertainty of the analysis, it is tempting to apply the transformation matrix $\mathbf{T}_{\mathrm{x}}$ to $\widehat{\mathbf{X}}^{\mathsf{f}}$, defined as the $N_{\mathrm{x}} \times N_{\mathrm{e}}$ matrix containing the N_{e} dominant modes of the augmented ensemble $\widehat{\mathbf{X}}$. Hence, we could propose the perturbation update

$$\widehat{\mathbf{X}}^{\mathsf{a}} = (\mathbf{T}_{\mathrm{x}})^{-1/2}\widehat{\mathbf{X}}^{\mathsf{f}}, \tag{8.6}$$

where the resulting analysis perturbation matrix $\widehat{\mathbf{X}}^{\mathsf{a}}$ is of size $N_{\mathrm{x}} \times N_{\mathrm{e}}$. It is remarkable that this perturbation update differs from equation (7.2b), the perturbation update which defines the LEnSRF algorithm. On the one hand, equation (7.2b) smoothly operates a transformation on the forecast perturbation matrix $\mathbf{X}^{\mathsf{f}}$ to obtain the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$, so that one would think that it could generate fewer imbalance compared to a transformation on the truncated modes $\widehat{\mathbf{X}}^{\mathsf{f}}$. On the other hand, the Frobenius norm of the difference between the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathsf{a}}$ and $\mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}$ is larger than the norm of the difference between $\bar{\mathbf{P}}^{\mathsf{a}}$ and $\widehat{\mathbf{X}}^{\mathsf{a}}(\widehat{\mathbf{X}}^{\mathsf{a}})^{\mathsf{T}}$, a fact which can also be checked numerically. Unfortunately, preliminary experiments using the L96 model and a modified LEnSRF algorithm, in which the perturbation update is defined by equation (8.6) instead of equation (7.2b), show that the update described by equation (8.6) is ineffective and systematically makes the filter diverge after a few assimilation cycles. This seems contradictory with the fact that this update captures as much uncertainty as possible, at least as measured using matrix norms.

The reason behind this apparent paradox is that in a cycled DA context based on equation (8.6), the localisation is essentially applied twice per cycle. Indeed, $\widehat{\mathbf{X}}^{\mathsf{f}}$ already captures the dominant contributions from a localised $\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{f}}$, hence a first footprint of localisation. The resulting $\widehat{\mathbf{X}}^{\mathsf{a}}$ would then form an analysis ensemble $\mathbf{E}^{\mathsf{a}}$ to be forecasted. In the next analysis step, the forecast statistics would be based on this forecast ensemble. The regularisation of the covariances would then require localisation, once again. Since localisation by Schur product is not idempotent[1] localisation is applied once too many. This is why equation (8.6) cannot be used to define a perturbation update in a cycled DA context.

This clarifies *a posteriori* why equation (7.2b) is well suited to define the perturbation update of the LEnSRF algorithm: localisation is applied only once in each assimilation cycle. This argument also implies that, when using CL, the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ should not be blindly identified with the modes carrying most of the uncertainty. However, it is tacitly hoped that the forecast of the analysis ensemble $\mathbf{E}^{\mathsf{a}}$ at the next assimilation cycle will be adequately regularised by the localisation matrix $\boldsymbol{\rho}$.

By contrast, the local perturbation updates in the LETKF algorithm, algorithm 2.4, are meant to capture most of the uncertainty within each local domain. Hence, the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ is representative of the main uncertainty modes in this case. However, even though the analysis ensemble $\mathbf{E}^{\mathsf{a}}$ obtained with the LETKF algorithm may better represent $\bar{\mathbf{P}}^{\mathsf{a}}$, this property could eventually fade away in the forecast because of the local validity of the analysis ensemble $\mathbf{E}^{\mathsf{a}}$. Incidentally, this suggests that the LETKF algorithm could be better suited for ensemble short-term forecast, which could be investigated in a future study. Numerical clues supporting this idea are nonetheless provided at the end of subsection 8.2.3.

8.1.2 Improving the consistency of the perturbation update

We have just seen that the widespread view on the local EnKF perturbation update which assumes a low-rank extraction $\mathbf{X}^{\mathsf{a}}$ from $\bar{\mathbf{P}}^{\mathsf{a}}$ with the hope that $\mathbf{X}^{\mathsf{a}}$ captures the most important directions of uncertainty, *i.e.*, $\mathbf{P}^{\mathsf{a}} \approx \mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}$, is only accurate for the LETKF algorithm. When using CL, the perturbations do not have to coincide with the dominant modes.

[1] Unless one uses a boxcar-like localisation matrix $\boldsymbol{\rho}$, which would not be a proper correlation matrix.

For the LEnSRF algorithm, we believe that it would be more consistent with how the perturbations are defined to look for a low-rank analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ such that

$$\bar{\mathbf{P}}^{\mathsf{a}} \approx \boldsymbol{\rho} \circ \left[\mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}\right], \tag{8.7}$$

instead of using equation (7.2b). Indeed, within equation (8.7), $\mathbf{X}^{\mathsf{a}}$ should not be interpreted as the dominant modes of $\bar{\mathbf{P}}^{\mathsf{a}}$ but as intermediate objects, perturbations whose short range covariances are indeed representative of the short range covariances of $\bar{\mathbf{P}}^{\mathsf{a}}$, but whose long range covariances are not used and possibly irrelevant. In the LEnSRF algorithm, the proper covariances will anyway be reconstructed with a Schur product after the forecast. A solution $\mathbf{X}^{\mathsf{a}}$ of equation (8.2) trades the accuracy of the representation of the long range covariances (which may eventually be discarded at the next assimilation cycle) for a potentially better accuracy of the short range covariances. Indeed, applying the localisation matrix $\boldsymbol{\rho}$ via the Schur product relaxes the long-range constraints and a better match with $\bar{\mathbf{P}}^{\mathsf{a}}$ can potentially be achieved for short range covariances.

Let $\mathcal{L}$ be the cost function defined as

$$\mathcal{L}(\mathbf{X}) \triangleq \ln\left\|\boldsymbol{\rho} \circ \left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}. \tag{8.8}$$

Our objective is to minimise $\mathcal{L}$ over all perturbation matrices $\mathbf{X}$ with N_{e} columns. As discussed in the following, this minimisation problem may have several solutions, which means that we will have to select one perturbation matrix $\mathbf{X}$ which minimises $\mathcal{L}$. The log-transformation applied to the norm is monotonically increasing and hence leaves the minima unchanged. This choice will be justified later on.

This problem is similar to the weighted low-rank approximation (WLRA) problem, which consists in minimising, for a given target matrix $\mathbf{V}$ and a given weighting matrix $\boldsymbol{\rho}$, the cost function $\mathcal{J}$ defined as

$$\mathcal{J}(\mathbf{A}) \triangleq \|\boldsymbol{\rho} \circ (\mathbf{A} - \mathbf{V})\|_{\mathrm{F}}, \tag{8.9}$$

over all matrices $\mathbf{A}$ of rank strictly smaller than N_{e} (Manton et al. 2003; Srebro and Jaakkola 2003). With the identification $\bar{\mathbf{P}}^{\mathsf{a}} = \boldsymbol{\rho} \circ \mathbf{V}$ and imposing the matrix $\mathbf{A}$ to be symmetric and positive semi-definite, the problem which consists in minimising $\mathcal{L}$ defined by equation (8.8) is seen to belong to the class of WLRA problems. As opposed to the uniform case, in which all elements of $\boldsymbol{\rho}$ are 1, and for which the minimiser simply coincides with the truncated svd of $\bar{\mathbf{P}}^{\mathsf{a}}$ by Eckart–Young theorem, the non-uniform case has no simple solution.[2]

Hence, we expect that the cost function $\mathcal{L}$ has no tractable minimiser. The literature of the WLRA problem focuses on the non-symmetric case, which would correspond for our problem to $\mathcal{L}(\mathbf{X}, \mathbf{Y}) = \ln\left\|\boldsymbol{\rho} \circ \left(\mathbf{X}\mathbf{Y}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}$. By contrast, our focus is on the symmetric case, which has less degrees of freedom. Still, it is unlikely to be amenable to a convex problem. Let us see why.

The cost function $\mathcal{L}$ is defined on the space of the perturbation matrices $\mathbf{X}$ with N_{e} columns, which is a convex subspace. Minimising $\mathcal{L}$ is equivalent to minimising $\left\|\boldsymbol{\rho} \circ \left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^{2}$ which is algebraic but nonetheless quartic in $\mathbf{X}$ and hence cannot be guaranteed to be convex. The problem is also equivalent to finding a matrix $\mathbf{P}$ of rank strictly smaller than N_{e} which

[2] It is actually known to be NP-hard.

Algorithm 8.1: Analysis step for the modified LEnSRF algorithm in the context of the GL system.

```
**Input:** $\mathbf{E}^{\mathsf{f}}\,[t_k]$, $\mathbf{y}\,[t_k]$
**Parameters:** $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{\rho}$
$\bar{\mathbf{x}} \leftarrow \mathbf{E}^{\mathsf{f}}\mathbf{1}/N_{\mathrm{e}}$
$\mathbf{X} \leftarrow \mathbf{E}^{\mathsf{f}}\big(\mathbf{I} - \mathbf{1}\mathbf{1}^{\mathsf{T}}/N_{\mathrm{e}}\big)/\sqrt{N_{\mathrm{e}}-1}$
$\mathbf{P} \leftarrow \boldsymbol{\rho} \circ \big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big)$
$\mathbf{K} \leftarrow \mathbf{P}\mathbf{H}^{\mathsf{T}}\big(\mathbf{H}\mathbf{P}\mathbf{H}^{\mathsf{T}} + \mathbf{R}\big)^{-1}$
$\bar{\mathbf{x}}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}} + \mathbf{K}\big(\mathbf{y} - \mathbf{H}\bar{\mathbf{x}}\big)$                // mean update
$\bar{\mathbf{P}}^{\mathsf{a}} \leftarrow (\mathbf{I} - \mathbf{K}\mathbf{H})\,\mathbf{P}$
$\mathbf{X}^{\mathsf{a}} \leftarrow \underset{\mathbf{Z}\in\mathbb{R}^{N_{\mathrm{x}}\times N_{\mathrm{e}}}}{\arg\min}\ \ln\Big\|\boldsymbol{\rho}\circ\big(\mathbf{Z}\mathbf{Z}^{\mathsf{T}}\big) - \bar{\mathbf{P}}^{\mathsf{a}}\Big\|_{\mathrm{F}}$    // new perturbation update
$\mathbf{E}^{\mathsf{a}} \leftarrow \bar{\mathbf{x}}^{\mathsf{a}}\mathbf{1}^{\mathsf{T}} + \sqrt{N_{\mathrm{e}}-1}\mathbf{X}^{\mathsf{a}}$
**Output:** $\mathbf{E}^{\mathsf{a}}\,[t_k]$
```

minimises $\big\|\boldsymbol{\rho}\circ\mathbf{P} - \bar{\mathbf{P}}^{\mathsf{a}}\big\|_{\mathrm{F}}^{2}$. This function is quadratic in $\mathbf{P}$. However, the space of the matrices $\mathbf{P}$ satisfying $\operatorname{rk}\mathbf{P} \leq N_{\mathrm{e}} - 1 < N_{\mathrm{x}}$ is not convex. Hence our problem may have several or even an infinite number of solutions (a variety). For instance, there are many redundant degrees of freedom such as $\mathcal{L}(\mathbf{X}) = \mathcal{L}(\mathbf{X}\mathbf{U})$ with $\mathbf{U}$ being any $N_{\mathrm{e}}\times N_{\mathrm{e}}$ orthogonal matrix. Therefore, the optimisation problem which consists in minimising $\mathcal{L}$ is degenerate. The modified LEnSRF algorithm with this new perturbation update is summarised in algorithm 8.1. In this algorithm, it is assumed that one minimiser of $\mathcal{L}$ is selected among all minimisers. Furthermore, the sample covariance matrix $\bar{\mathbf{P}}$ and the perturbation matrix $\mathbf{X}$ are related by

$$\bar{\mathbf{P}} \approx \boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big), \tag{8.10}$$

throughout the entire assimilation cycle. In other words, equation (8.10) is valid for both the forecast and the analysis quantities.

With a view to efficiently minimising the cost function $\mathcal{L}$, let us compute its gradient. The variation of $\mathcal{L}(\mathbf{X})$ with respect to a variation of $\mathbf{X}$ is

$$\delta\mathcal{L}(\mathbf{X}) = \frac{\delta\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}{2\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}, \tag{8.11}$$

$$= \frac{\delta\,\mathrm{tr}\big[\boldsymbol{\Delta}\boldsymbol{\Delta}^{\mathsf{T}}\big]}{2\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}, \tag{8.12}$$

$$= \frac{\mathrm{tr}\Big[\boldsymbol{\rho}\circ\big[(\delta\mathbf{X})\mathbf{X}^{\mathsf{T}}\big]\boldsymbol{\Delta} + \boldsymbol{\rho}\circ\big[\mathbf{X}(\delta\mathbf{X})^{\mathsf{T}}\big]\boldsymbol{\Delta}\Big]}{\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}, \tag{8.13}$$

where $\boldsymbol{\Delta} \triangleq \boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big) - \bar{\mathbf{P}}^{\mathrm{a}}$. Now, using the identity

$$\mathrm{tr}\big[(\mathbf{A}\circ\mathbf{B})\,\mathbf{C}\big] = \mathrm{tr}\big[\mathbf{A}\,(\mathbf{B}^{\mathsf{T}}\circ\mathbf{C})\big], \tag{8.14}$$

which is valid for any matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ with compatible size, we obtain

$$\delta\mathcal{L}(\mathbf{X}) = \frac{2\,\mathrm{tr}\Big[(\delta\mathbf{X})^{\mathsf{T}}(\boldsymbol{\rho}\circ\boldsymbol{\Delta})\mathbf{X}\Big]}{\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}, \tag{8.15}$$

which yields the matrix gradient

$$\nabla\mathcal{L}(\mathbf{X}) = \frac{2\,(\boldsymbol{\rho}\circ\boldsymbol{\Delta})\,\mathbf{X}}{\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2}, \tag{8.16}$$

in other words, the gradient of $\mathcal{L}(\mathbf{X})$ with respect to each component of the perturbation matrix $\mathbf{X}$. When implementing the new LEnSRF algorithm, we provide the gradient $\nabla\mathcal{L}$ as well as the value of the cost function $\mathcal{L}$ to an off-the-shelf numerical optimisation algorithm, such as the L-BFGS-B algorithm (Byrd et al. 1995). The cost function $\mathcal{L}$ may not only have many global minima, but it may also have many local minima. As a consequence it may not be possible to find a global minimum with the L-BFGS-B algorithm.

8.1.3 Parametrised minimisation

Instead of minimising $\mathcal{L}$ over $\mathbf{X}$ which has redundant degrees of freedom, we use an RQ decomposition of the perturbation matrix $\mathbf{X}$, which is obtained from a QR decomposition (Golub and Van Loan 2013) of $\mathbf{X}^{\mathsf{T}}$ as

$$\mathbf{X} = \boldsymbol{\Omega}\mathbf{U}, \tag{8.17}$$

where $\mathbf{U}$ is an $N_{\mathrm{e}}\times N_{\mathrm{e}}$ orthogonal matrix and $\boldsymbol{\Omega}$ is an $N_{\mathrm{x}}\times N_{\mathrm{e}}$ lower triangular (actually trapezoidal) matrix. Hence, $\mathbf{X}\mathbf{X}^{\mathsf{T}} = \boldsymbol{\Omega}\boldsymbol{\Omega}^{\mathsf{T}}$ only depends on $\boldsymbol{\Omega}$. The number of degrees of freedom in this parametrisation is that of $\boldsymbol{\Omega}$, which is

$$N_{\mathrm{e}}N_{\mathrm{x}} - N_{\mathrm{e}}\frac{N_{\mathrm{e}}-1}{2} = N_{\mathrm{e}}(N_{\mathrm{x}}-N_{\mathrm{e}}) + N_{\mathrm{e}}\frac{N_{\mathrm{e}}+1}{2}. \tag{8.18}$$

A parametrised minimisation can easily be implemented using the cost function

$$\mathcal{L}(\boldsymbol{\Omega}) = \ln\left\|\boldsymbol{\rho} \circ \left(\boldsymbol{\Omega}\boldsymbol{\Omega}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}, \tag{8.19}$$

and its gradient

$$\nabla\mathcal{L}(\boldsymbol{\Omega}) = \frac{2\,\boldsymbol{\Pi}_{\boldsymbol{\Omega}}(\boldsymbol{\rho} \circ \boldsymbol{\Delta})\,\boldsymbol{\Omega}}{\|\boldsymbol{\Delta}\|_{\mathrm{F}}^{2}}, \tag{8.20}$$

where $\boldsymbol{\Pi}_{\boldsymbol{\Omega}}$ is the projector which sets to zero the upper triangular part of $(\boldsymbol{\rho} \circ \boldsymbol{\Delta})\,\boldsymbol{\Omega}$ as it is in $\boldsymbol{\Omega}$.

In the numerical experiments of section 8.2, we use this parametrised minimisation. However, the plain method using the non-parametrised minimisation works as well, although there is no guarantee to find the same local minimum because of the potential non-convexity of $\mathcal{L}$.

In subsection 8.2.4, we address the question of the matrix norm choice in the definition of the cost function $\mathcal{L}$ with equation (8.8). In particular, we test the use of the spectral and nuclear matrix norms, and, more generally, of the Schatten p-norms. We found that these choices did not make much of a difference but that the choice of either the spectral or the nuclear norm, at the ends of the Schatten range, could lead to inaccurate numerical results.

Finally, coming back to the definition of the cost function $\mathcal{L}$, we have chosen to apply a log-transformation to the norm to level off the ups and downs of the function. Since the functions are non-convex, a quasi-Newton minimiser such as the L-BFGS-B algorithm may behave differently in terms of convergence and local minima depending on the nature of the transformation. Hence, the log-transformation should not be considered totally innocuous. In practice, we found using the log-transformation systematically beneficial.

8.1.4 The forecast step

Because we have offered a novel view on the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$ and how it is computed in the analysis step, we now need to examine how the forecast step is affected by this change of standpoint. If not, there would be a risk of breaking the consistency during the forecast step.

As previously explained in subsection 8.1.1, an asset of the LETKF algorithm is that $\mathbf{X}^{\mathsf{a}}$ represent the dominant modes of $\bar{\mathbf{P}}^{\mathsf{a}}$. Hence, the forecast uncertainty must be approximated by the forecast of these modes. Nonetheless, by construction, the statistics of these modes before or after the forecast are only valid on local domains, *i.e.*, for short spatial separations. By contrast, with the modified LEnSRF algorithm, recognising that

$$\bar{\mathbf{P}}^{\mathsf{a}} \approx \boldsymbol{\rho} \circ \left[\mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}\right] \tag{8.21}$$

makes forecasting more intricate. Indeed, this representation is meant to model statistics valid for larger spatial separations. How would one forecast such a representation of $\bar{\mathbf{P}}^{\mathsf{a}}$?

A practical answer to this problem has been proposed by Bocquet (2016) in a linear and deterministic context, *i.e.*, when the dynamical model $\mathbf{M}$ is linear (which is the case in the GL system) and when there is no model error. First, $\mathbf{X}^{\mathsf{a}}$ is assumed to represent N_{e} genuine physical perturbations, which are forecasted using the model $\mathbf{M}$ from t_k to t_{k+1}:

$$\mathbf{X}^{\mathsf{f}}_{k+1} = \mathbf{M}\mathbf{X}^{\mathsf{a}}_{k}. \tag{8.22}$$

Second, the localisation matrix $\boldsymbol{\rho}$ should be time-dependent and satisfy, in the time continuum limit, the Liouville equation

$$\frac{\partial\,\mathrm{vec}(\boldsymbol{\rho})}{\partial t} = \left[\mathbf{M}\otimes\mathbf{I}+\mathbf{I}\otimes\mathbf{M}\,,\,\mathrm{vec}(\boldsymbol{\rho})\right]. \tag{8.23}$$

In this equation, $\mathrm{vec}(\boldsymbol{\rho})$ is the vectorised localisation matrix $\boldsymbol{\rho}$, that is the vector whose N_{x}^2 elements are those of $\boldsymbol{\rho}$, and $\otimes$ denotes the Kronecker product between two copies of the state space $\mathbb{R}^{N_{\mathrm{x}}}$.

In the case where the dynamics can be approximated as hyperbolic, and in the limit where space is continuous, a closed-form equation can be obtained for $\rho(x_1, x_2, t)$ (see equation (A14) of Bocquet 2016). If diffusion is present, there is no such closed-form equation. See also Kalnay et al. (2012) and Desroziers et al. (2016) who have considered this issue in other contexts.

The key point is that in practice and for moderate forecast lead times, the localisation matrix $\boldsymbol{\rho}$ can roughly be assumed to be static. This is what will be used in the numerical experiments of section 8.2. When the time interval between consecutive observation Δt is larger, one could assume at the next order approximation that the localisation length used to obtain the prior at time t_{k+1} is larger than the one used in the analysis at time t_k, because of an effective diffusion either generated by genuine diffusion or by averaged mixing advection (as stressed in the appendix A of Bocquet 2016).

In conclusion, if $\bar{\mathbf{P}}^{\mathsf{a}}$ is approximated by $\boldsymbol{\rho}\circ\left[\mathbf{X}^{\mathsf{a}}(\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}\right]$, then $\boldsymbol{\rho}\circ\left[\mathbf{M}\mathbf{X}^{\mathsf{a}}(\mathbf{M}\mathbf{X}^{\mathsf{a}})^{\mathsf{T}}\right]$ is an acceptable approximation of $\mathbf{M}\bar{\mathbf{P}}^{\mathsf{a}}\mathbf{M}^{\mathsf{T}}$, the forecast error covariance matrix at the next cycle. In other words, the forecast step does not need to be modified.

8.1.5 Algorithmic complexity of computing the cost function and its gradient

In this subsection, we analyse the algorithmic complexity of computing the cost function $\mathcal{L}$ and its gradient. Indeed, both would be required by a quasi-Newton minimiser and both involve the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathsf{a}}$.

8.1.5.1 Bottlenecks in the method

The cost function $\mathcal{L}$ requires computing

$$\begin{aligned}
\|\boldsymbol{\Delta}\|_{\mathrm{F}}^2 &= \left\|\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^2, && (8.24)\\
&= \left\|\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)\right\|_{\mathrm{F}}^2 + \left\|\bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^2 - 2\,\mathrm{tr}\left[\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)\bar{\mathbf{P}}^{\mathsf{a}}\right], && (8.25)\\
&= \mathrm{tr}\Big[\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right)\left[\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - 2\bar{\mathbf{P}}^{\mathsf{a}}\right]\Big] + \left\|\bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^2, && (8.26)\\
&= \mathrm{tr}\Big[\mathbf{X}\mathbf{X}^{\mathsf{T}}\boldsymbol{\rho}\circ\left[\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - 2\bar{\mathbf{P}}^{\mathsf{a}}\right]\Big] + \left\|\bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^2, && (8.27)\\
&= \mathrm{tr}\Big[\mathbf{X}^{\mathsf{T}}\boldsymbol{\rho}\circ\left[\boldsymbol{\rho}\circ\left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - 2\bar{\mathbf{P}}^{\mathsf{a}}\right]\mathbf{X}\Big] + \left\|\bar{\mathbf{P}}^{\mathsf{a}}\right\|_{\mathrm{F}}^2. && (8.28)
\end{aligned}$$

Moreover, the gradient of $\mathcal{L}$, given by equation (8.16), can be written

$$\nabla\mathcal{L}(\mathbf{X}) = \frac{2\big[\boldsymbol{\rho}\circ\boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big)\mathbf{X} - \big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{a}}\big)\mathbf{X}\big]}{\|\boldsymbol{\Delta}\|_{\mathrm{F}}^{2}}, \tag{8.29}$$

in which the normalisation factor is precisely le left-hand side of equation (8.24).

In conclusion, for the computation of both the cost function $\mathcal{L}$ and its gradient, we need to evaluate a first term in the form $\boldsymbol{\rho}\circ\boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big)\mathbf{X}$ and a second term in the form $\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{a}}\big)\mathbf{X}$.

8.1.5.2 Efficient evaluation of the first term

Using the factorisation property stated by equation (7.31), we conclude that the algorithmic complexity of computing the first term $\boldsymbol{\rho}\circ\boldsymbol{\rho}\circ\big(\mathbf{X}\mathbf{X}^{\mathsf{T}}\big)\mathbf{X}$ is

- $\mathcal{O}\big(N_{\mathrm{e}}^{2}N_{\mathrm{x}}N_{\mathrm{b}}\big)$ if $\boldsymbol{\rho}$ is banded with non-zero elements on the main N_{b} diagonals;
- $\mathcal{O}\big(N_{\mathrm{e}}^{2}N_{\mathrm{x}}\ln N_{\mathrm{x}}\big)$ if $\boldsymbol{\rho}$ is circulant.

8.1.5.3 Efficient evaluation of the second term

Assuming that the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathsf{a}}$ is entirely known, for all vector $\mathbf{v}\in\mathbb{R}^{N_{\mathrm{x}}}$, the n-th element of $\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{a}}\big)\,\mathbf{v}$ is given by

$$\Big[\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{a}}\big)\,\mathbf{v}\Big]_{n} = \sum_{m=1}^{N_{\mathrm{x}}}\big[\boldsymbol{\rho}\big]_{n,m}\big[\bar{\mathbf{P}}^{\mathsf{a}}\big]_{n,m}\big[\mathbf{v}\big]_{m}, \tag{8.30}$$

$$= \sum_{m=1}^{N_{\mathrm{x}}}\big[\bar{\mathbf{P}}^{\mathsf{a}}\big]_{n,m}\big[\boldsymbol{\rho}_{n}\circ\mathbf{v}\big]_{m}, \tag{8.31}$$

$$= \bar{\mathbf{P}}^{\mathsf{a}}_{n}(\boldsymbol{\rho}_{n}\circ\mathbf{v}), \tag{8.32}$$

where $\boldsymbol{\rho}_{n}$ is the n-th column of the localisation matrix $\boldsymbol{\rho}$ and $\bar{\mathbf{P}}^{\mathsf{a}}_{n}$ is the n-th row of the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathsf{a}}$.

As a consequence, the algorithmic complexity of computing the second term $\big(\boldsymbol{\rho}\circ\bar{\mathbf{P}}^{\mathsf{a}}\big)\,\mathbf{X}$ is $\mathcal{O}(N_{\mathrm{x}}N_{\mathrm{b}}N_{\mathrm{e}})$ if the localisation matrix $\boldsymbol{\rho}$ is banded with non-zero elements on the main N_{b} diagonals. This cost is acceptable, in other words it does not departs much from $\mathcal{O}(N_{\mathrm{x}})$. However, it does not account for the cost of evaluating $\bar{\mathbf{P}}^{\mathsf{a}}$.

Suppose that we have an approximate factorisation of $\bar{\mathbf{P}}^{\mathsf{a}}$ under the form $\bar{\mathbf{P}}^{\mathsf{a}}\approx\widehat{\mathbf{X}}^{\mathsf{a}}\widehat{\mathbf{X}}^{\mathsf{a}}$, where $\widehat{\mathbf{X}}$ is an augmented ensemble of size $\widehat{N}_{\mathrm{e}}$. Such a factorisation can be obtained, for example, by using the methods presented in section 7.2. With this factorisation, the second term becomes $\boldsymbol{\rho}\circ\big[\widehat{\mathbf{X}}^{\mathsf{a}}(\widehat{\mathbf{X}}^{\mathsf{a}})^{\mathsf{T}}\big]\,\mathbf{X}$, which is very similar to the first term. Therefore, we conclude that the algorithmic complexity of computing the second term is

- $\mathcal{O}\big(N_{\mathrm{e}}\widehat{N}_{\mathrm{e}}N_{\mathrm{x}}N_{\mathrm{b}}\big)$ if $\boldsymbol{\rho}$ is banded with non-zero elements on the main N_{b} diagonals;
- $\mathcal{O}\big(N_{\mathrm{e}}\widehat{N}_{\mathrm{e}}N_{\mathrm{x}}\ln N_{\mathrm{x}}\big)$ if $\boldsymbol{\rho}$ is circulant.

In this case, as well as for the evaluation of the first term, the computations can be easily parallelised, reducing the algorithmic complexity by a factor N_t, the number of threads running in parallel.

For completeness, it is interesting to remark that, if the observations are assumed to be local, then the main diagonals of $\bar{\mathbf{P}}^\mathsf{a}$ can be computed using local approximations, in a way similar to the strategy followed by the LETKF algorithm. Indeed, with the LETKF algorithm the n-th local analysis implies

$$\mathbf{X}^\mathsf{a}\big(\mathbf{X}^\mathsf{a}\big)^\mathsf{T} = \mathbf{X}^\mathsf{f}\big(\mathbf{I} + \mathbf{Y}^\mathsf{T}\mathbf{R}_n^{-1}\mathbf{Y}\big)^{-1}\big(\mathbf{X}^\mathsf{f}\big)^\mathsf{T}, \tag{8.33}$$

where $\mathbf{R}_n$ is the tapered observation error covariance matrix. From $\mathbf{X}^\mathsf{a}\big(\mathbf{X}^\mathsf{a}\big)^\mathsf{T}$, one would typically extract the n-th column to form the n-th column of the (global) $\bar{\mathbf{P}}^\mathsf{a}$. If the localisation matrix $\boldsymbol{\rho}$ is banded with non-zero elements on the main N_b diagonals, then the algorithmic complexity of obtaining the N_b relevant elements of one column of $\bar{\mathbf{P}}^\mathsf{a}$ is typically[3] $\mathcal{O}\big((N_\mathrm{y}^\ell)^2 N_\mathrm{e} + N_\mathrm{b}^2 N_\mathrm{e}\big)$, where the first term corresponds to the computation of the local transformation matrix $\mathbf{T}_\mathrm{e}$ and the second term correspond to the application of this $\mathbf{T}_\mathrm{e}$. Finally, the algorithmic complexity of evaluating all columns is $\mathcal{O}\big(N_\mathrm{x}(N_\mathrm{y}^\ell)^2 N_\mathrm{e} + N_\mathrm{x} N_\mathrm{b}^2 N_\mathrm{e}\big)$, which, again, can be reduced by a factor N_t, the number of threads running in parallel.

Of course, one of the primary reasons for using CL is its ability to assimilate non-local observations. Hence, the assumption of locality made here defeats one of the key purpose of using CL. Nonetheless, we shall see that even with local observations, the new perturbation update method developed in subsection 8.1.2 can be beneficial.

8.2 Numerical experiments

8.2.1 Properties of the new perturbations

At first, we are interested in comparing the shape of the perturbations obtained with the original method and with the new method. We also wish to explore how much the cost function $\mathcal{L}$ can be rendered small. To that end, we consider a variant of the one-dimensional model introduced in paragraph 7.3.1.1.

8.2.1.1 A simple one-dimensional model

Using the notation introduced in paragraph 7.3.1.1, the reference covariance matrix $\mathbf{P}_\mathrm{ref}$ is constructed as the matrix whose correlation structure is $\mathbf{C}(\ell_\mathrm{ref})$ and whose standard deviation vector $\mathbf{r}$ is a random draw from the distribution $\mathcal{LN}\big[\mathbf{0}, \alpha_\mathrm{c}\,\mathbf{C}(\ell_\mathrm{c})\big]$, where α_c, ℓ_ref and ℓ_c are three parameters to be determined. In other words, we have

$$\mathbf{P}_\mathrm{ref} \triangleq \mathbf{D}(\mathbf{r})\,\mathbf{C}(\ell_\mathrm{ref})\,\mathbf{D}(\mathbf{r}). \tag{8.34}$$

For this experiment, we use $N_\mathrm{x} = 400$ state variables, and the following values for the parameters: $\alpha_\mathrm{c} = 0.3$, $\ell_\mathrm{c} = 13$ grid points, and $\ell_\mathrm{ref} = 10$ grid points. The resulting reference covariance matrix $\mathbf{P}_\mathrm{ref}$ is displayed in figure 8.1.

[3] For simplicity, it is assumed here that $N_\mathrm{y}^\ell \geq N_\mathrm{e}$ and $N_\mathrm{b} \geq N_\mathrm{e}$.

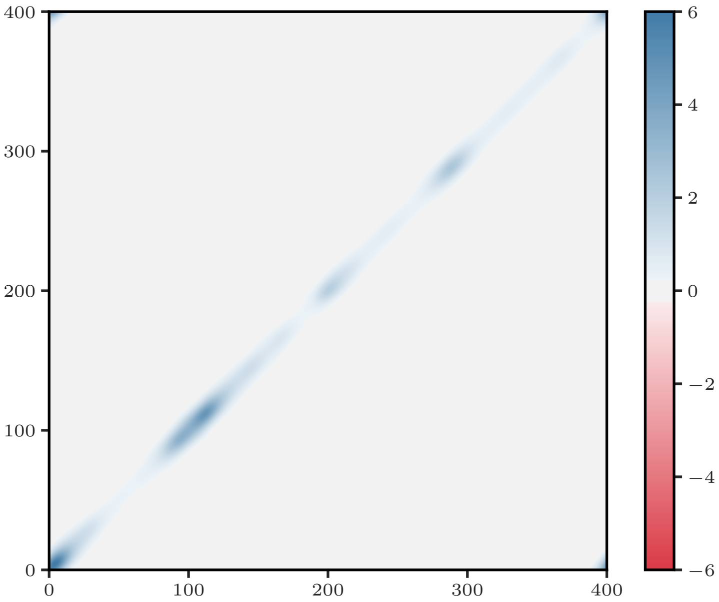

Figure 8.1: Reference covariance matrix $\mathbf{P}_{\mathrm{ref}}$.

We now compare different approximations of $\mathbf{P}_{\mathrm{ref}}$ using an ensemble of $N_{\mathrm{e}} = 8$ members.

1. Let $\bar{\mathbf{X}}$ be the perturbation matrix of an ensemble of N_{e} members independently drawn from the distribution $\mathcal{N}[\mathbf{0}, \mathbf{P}_{\mathrm{ref}}]$.
2. Let $\widehat{\mathbf{X}}$ be the perturbation matrix associated to the main N_{e} modes of $\mathbf{P}_{\mathrm{ref}}$.
3. Let $\bar{\mathbf{X}}^*$ be the perturbation matrix obtained using the new method, that is by minimising the cost function $\mathbf{X} \mapsto \ln\left\|\boldsymbol{\rho} \circ \left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \mathbf{P}_{\mathrm{ref}}\right\|_{\mathrm{F}}$, starting the minimisation from $\bar{\mathbf{X}}$.
4. Finally, let $\widehat{\mathbf{X}}^*$ be the perturbation matrix obtained using the new method, where the starting point of the minimisation is now chosen to be $\widehat{\mathbf{X}}$.

In all cases, the sample covariance matrices $\bar{\mathbf{P}} \triangleq \bar{\mathbf{X}}\bar{\mathbf{X}}^{\mathsf{T}}$, $\widehat{\mathbf{P}} \triangleq \widehat{\mathbf{X}}\widehat{\mathbf{X}}^{\mathsf{T}}$, $\bar{\mathbf{P}}^* \triangleq \bar{\mathbf{X}}^*(\bar{\mathbf{X}}^*)^{\mathsf{T}}$, and $\widehat{\mathbf{P}}^* \triangleq \widehat{\mathbf{X}}^*(\widehat{\mathbf{X}}^*)^{\mathsf{T}}$ may be localised using the localisation matrix $\boldsymbol{\rho} = \mathbf{C}(\ell_{\mathrm{ref}})$.

8.2.1.2 Results and discussion

The sample covariance matrices $\bar{\mathbf{P}}$, $\widehat{\mathbf{P}}$, and $\widehat{\mathbf{P}}^*$ as well as their localised counterparts $\boldsymbol{\rho} \circ \bar{\mathbf{P}}$, $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}$, and $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}^*$ are displayed in figure 8.2. Furthermore, figure 8.3 displays the perturbations, *i.e.*, the columns of the perturbation matrices, in all four cases.

It is clear from figure 8.2 that $\widehat{\mathbf{P}}^*$ seems unphysical with rather long-range correlations, but that its localised counterpart $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}^*$ is, as a result of its construction, a remarkably close match to $\mathbf{P}_{\mathrm{ref}}$. The sample covariance matrix $\widehat{\mathbf{P}}$ seems a rather good approximation of $\mathbf{P}_{\mathrm{ref}}$. However, it is clear that its localised counterpart $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}$ has a thinner structure along

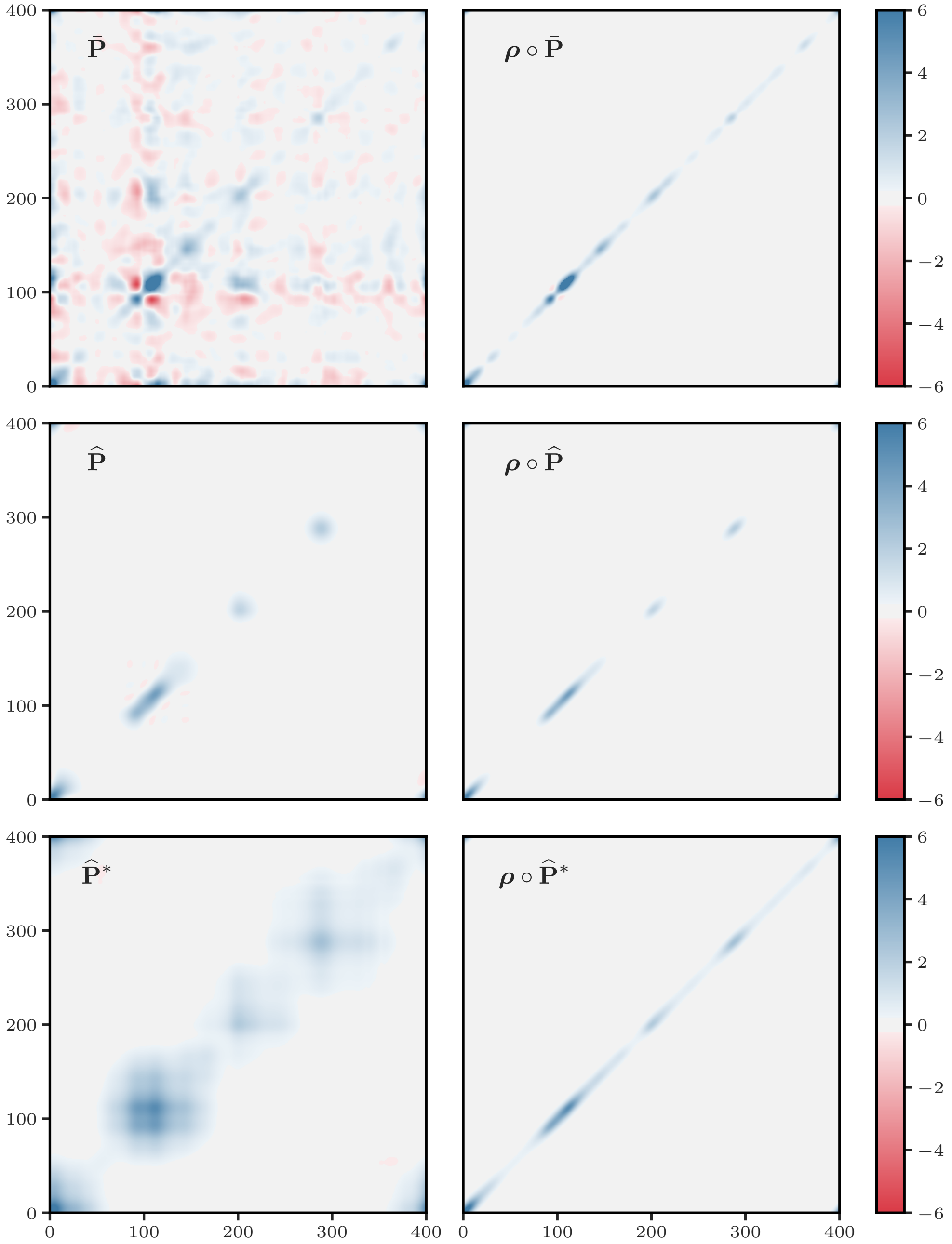

Figure 8.2: Sample covariance matrices $\bar{\mathbf{P}}$ (top-left panel), $\widehat{\mathbf{P}}$ (central-left panel), and $\widehat{\mathbf{P}}^*$ (bottom-left panel), as well as their localised counterparts $\boldsymbol{\rho} \circ \bar{\mathbf{P}}$ (top-right panel), $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}$ (central-right panel), and $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}^*$ (bottom-right panel). The reference covariance matrix $\mathbf{P}_{\text{ref}}$, displayed in figure 8.1, is visually very close $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}^*$.

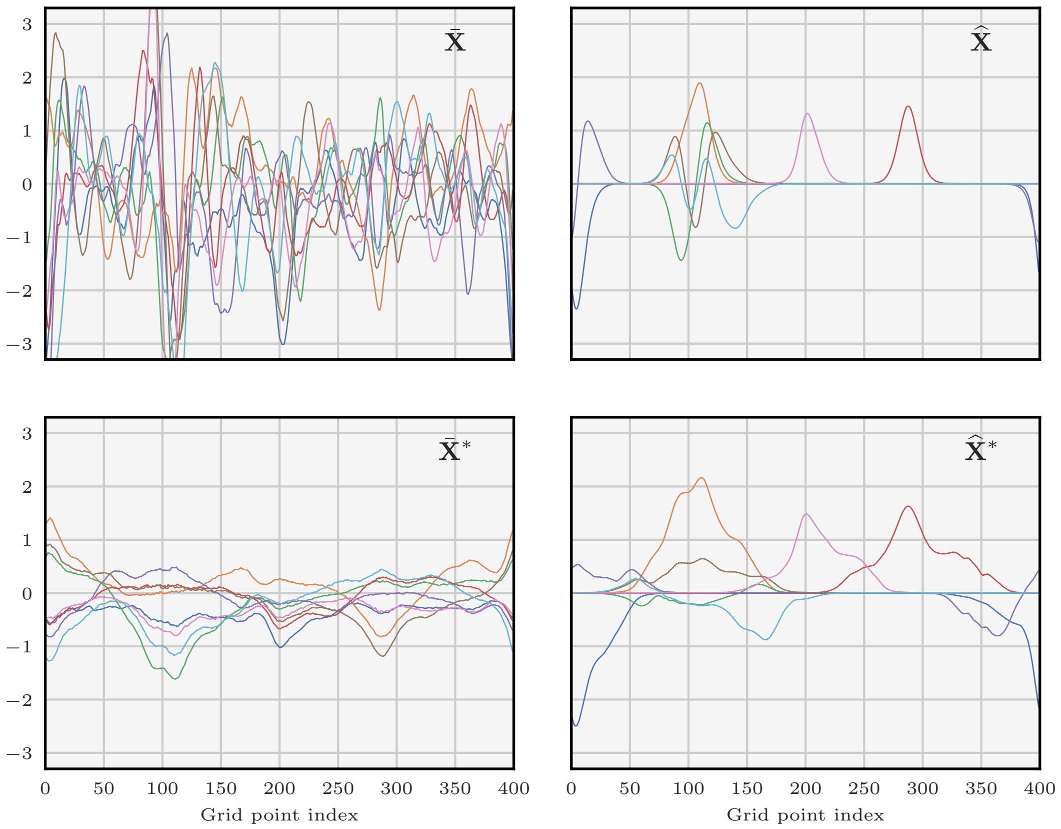

Figure 8.3: Ensemble of $N_\mathrm{e} = 8$ perturbations as a function of the grid point index for the perturbation matrices $\bar{\mathbf{X}}$ (top-left panel), $\widehat{\mathbf{X}}$ (top-right panel), $\bar{\mathbf{X}}^*$ (bottom-left panel), and $\widehat{\mathbf{X}}^*$ (bottom-right panel).

Table 8.1: Averaged Frobenius norm between the reference covariance matrix $\mathbf{P}_{\text{ref}}$ and the sample covariance matrices $\bar{\mathbf{P}}$, $\widehat{\mathbf{P}}$, $\widehat{\mathbf{P}}^*$, and $\bar{\mathbf{P}}^*$ (first row), as well as their localised counterparts (second row). For the sake of comparison note that, on average, $\|\mathbf{P}_{\text{ref}}\|_{\text{F}} = 87$.

Norm	$\bar{\mathbf{P}}$	$\widehat{\mathbf{P}}$	$\widehat{\mathbf{P}}^*$	$\bar{\mathbf{P}}^*$
$\|\mathbf{P}_{\text{ref}} - \mathbf{P}\|_{\text{F}}$	194	50	331	335
$\|\mathbf{P}_{\text{ref}} - \boldsymbol{\rho} \circ \mathbf{P}\|_{\text{F}}$	49	49	0.05	0.06

the diagonal than $\mathbf{P}_{\text{ref}}$, which can be seen as a manifestation of the double application of localisation. These visual impressions on a single realisation are confirmed by computing the Frobenius norm of the difference between $\mathbf{P}_{\text{ref}}$ and either the sample covariance matrix or its localised counterpart. The norm is averaged over 10^3 realisations of the whole experiment, and the results are reported in table 8.1. In particular, $\boldsymbol{\rho} \circ \widehat{\mathbf{P}}^*$ and $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^*$ are both close match to $\mathbf{P}_{\text{ref}}$, and their residual discrepancy to $\mathbf{P}_{\text{ref}}$, as measured by the Frobenius matrix norm, are very small and similar, though not identical.

As seen in figure 8.3, the perturbations in $\widehat{\mathbf{X}}$ are rather local and peaked functions, which could be expected since they represent the first main modes of $\mathbf{P}_{\text{ref}}$. The perturbations in $\widehat{\mathbf{X}}^*$, obtained with the new method starting the minimisation from $\widehat{\mathbf{X}}$ are much broader functions with a larger support. This is due to the weaker constraints imposed on these perturbations. However, they remain partially localised, meaning that they partly vanish on the domain. The perturbations in $\bar{\mathbf{X}}^*$, obtained with the new method but starting the minimisation from $\bar{\mathbf{X}}$ are also broad functions. However, as opposed to the perturbations in $\widehat{\mathbf{X}}^*$, they do not partially vanish, and are barely local. This shows that the cost function $\mathcal{L}$ indeed has a set of potentially distinct minimisers and that the solution to which the minimisation converges captures traits of the starting perturbation matrix.

8.2.2 Accuracy of the modified LEnSRF algorithm

In this subsection, the performance of the modified LEnSRF algorithm is illustrated using twin experiments of the L96 model described in subsection 5.1.2, and of the Kuramoto–Sivashinsky (KS) model, described in the following paragraph.

For these experiments, both the L96 and KS models are used in a mildly nonlinear configuration. For the L96 model, the mildly nonlinear configuration is presented in paragraph 5.1.2.2, and for the KS model it is presented in the following paragraph.

8.2.2.1 The Kuramoto–Sivashinsky model

The KS model (Kuramoto and Tsuzuki 1975, 1976; Sivashinsky 1977) is a low-order one-dimensional chaotic model whose evolution is given by the following partial differential equation (PDE):

$$\frac{\partial u}{\partial t} = -u\frac{\partial u}{\partial x} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^4 u}{\partial x^4}, \tag{8.35}$$

over the domain $x \in [0, 32\pi]$. As opposed to the L96 model, the KS model is continuous though numerically discretised in Fourier modes. It is characterised by sharp density gradients so that it may be expected that local EnKF algorithms are prone to imbalance. The model is integrated using the ETDRK4 method (Kassam and Trefethen 2005) with an integration time step δt equal to 0.5 unit of time, and without model error.

For the KS model, we define a *mildly nonlinear* DA configuration as follows. The domain $[0, 32\pi]$ is discretised using $N_{\mathrm{x}} = 128$ Fourier modes, which corresponds to $N_{\mathrm{x}} = 128$ collocation grid points. The time interval between consecutive observations Δt is set to 1 unit of time, and the observation vector $\mathbf{y}$ is computed from the truth $\mathbf{x}^{\mathrm{t}}$ using

$$\mathbf{y} = \mathbf{x}^{\mathrm{t}} + \mathbf{e}^{\mathrm{o}}, \quad \mathbf{e}^{\mathrm{o}} \sim \mathcal{N}[\mathbf{0}, \mathbf{I}]. \tag{8.36}$$

In this configuration, the number of unstable and neutral modes of the dynamics is 14.

8.2.2.2 Implementation notes

In these experiments, three algorithms are compared:

1. the LETKF algorithm, algorithm 2.4;
2. the LEnSRF algorithm, algorithm 7.1, in which the transformation matrix $\mathbf{T}_{\mathrm{x}}$ is computed exactly in these low-order DA systems;
3. the modified LEnSRF algorithm, algorithm 8.1, in which the estimated analysis error covariance matrix $\bar{\mathbf{P}}^{\mathrm{a}}$ is computed exactly in these low-order DA systems, and in which the starting point for the minimisation is the forecast perturbation matrix $\mathbf{X}^{\mathrm{f}}$, the natural incremental standpoint.

The performance criterion is the RMSE score described in subsection 5.1.1. In order to ensure the convergence of the statistical indicators, we use a spin-up period of $N_{\mathrm{s}} = 2 \times 10^3$ assimilation cycles and a total simulation period of at least $N_{\mathrm{c}} = N_{\mathrm{s}} + 2 \times 10^4$ assimilation cycles. Furthermore, each experiment is performed 10 times and the scores are averaged over the 10 realisations.

When the ensemble size N_{e} is smaller than the number of unstable and neutral modes of the dynamics (which is 14 for both models), localisation is mandatory to avoid the divergence of the algorithms. As in the experiments of subsection 7.3.2, the localisation matrix $\boldsymbol{\rho}$ is constructed as $\mathbf{C}(\ell)$, where ℓ is the localisation radius.

As presented in subsection 2.5.2, in order to mitigate the sampling errors, multiplicative inflation is used after the analysis step with a fixed multiplicative inflation factor λ. When showing the evolution of the RMSE score as a function of the multiplicative inflation factor λ, the localisation radius ℓ is optimally tuned to yield the lowest RMSE score. When showing the evolution of the RMSE score as a function of the ensemble size N_{e}, both the multiplicative inflation factor λ and the localisation radius ℓ are optimally tuned to yield the lowest RMSE score.

Finally, as presented in paragraph 2.3.2.3, random rotations are applied after each analysis step. It does marginally improve the RMSE scores for large values of the ensemble size N_{e}.

8.2.2.3 Results

Figure 8.4 shows the evolution of the RMSE score, of the optimal multiplicative inflation factor λ, and of the optimal localisation radius ℓ as a function of the ensemble size N_e. Furthermore, figure 8.5 shows the evolution of the RMSE score as a function of the multiplicative inflation factor λ. Let us first consider the results for the L96 model.

First, the LETKF and the LEnSRF algorithms yield similar RMSE scores and optimal λ for all values of the ensemble size N_e, but the LETKF algorithm has the edge for both the RMSE score and the multiplicative inflation factor. The optimal ℓ for all three algorithms are similar, in particular thanks to the approximate correspondence between the way $\mathbf{R}^{-1}$ is tapered in the LETKF algorithm (as presented in paragraph 2.5.4.1) and the way $\bar{\mathbf{P}}^\mathsf{f}$ is localised in the LEnSRF algorithm (with $\boldsymbol{\rho} \circ \bar{\mathbf{P}}^\mathsf{f}$). Nonetheless the optimal ℓ of the LEnSRF algorithm is smaller than that of the other two algorithms, especially for larger ensembles.

Second, the modified LEnSRF algorithm yield lower RMSE scores and significantly lower optimal λ than the other two algorithms. The improvement in RMSE score is in the range $3\% - 6\%$, which is significant in these very well-tuned and documented DA configurations, where such gain is very difficult to obtain.

The evolution of the RMSE score as a function of λ shows that the requirement for inflation of the modified LEnSRF algorithm is actually very small. For an ensemble of $N_\mathrm{e} = 8$ and 16 members, inflation is barely needed. In the extreme case of an ensemble of $N_\mathrm{e} = 4$ members, the modified LEnSRF algorithm does show a need for inflation, but much smaller than that of the other two algorithms. This points to the robustness of the modified LEnSRF algorithm.

By construction, the localised analysis sample covariance matrix $\boldsymbol{\rho} \circ \left[\mathbf{X}^\mathsf{a}(\mathbf{X}^\mathsf{a})^\mathsf{T}\right]$, is a better match to $\bar{\mathbf{P}}^\mathsf{a}$ when the analysis perturbation matrix $\mathbf{X}^\mathsf{a}$ is obtained with the new method (as in the modified LEnSRF algorithm) than when it is obtained using equation (7.2b) (as in the original LEnSRF algorithm). This might explain the lesser requirement for inflation.

We speculate that this lesser need for inflation in the modified LEnSRF algorithm may also be interpreted as a reduced imbalance in the analysis perturbation matrix $\mathbf{X}^\mathsf{a}$. If true, this implies that for the L96 model in this mildly nonlinear configuration, the residual inflation required in the LETKF and LEnSRF algorithms does not so much originate from the sampling errors but from the imbalance generated by localisation. This, however, can only be validated on physically more complex, two- or three-dimensional models.

The results for the KS model are very similar to those for the L96 model. The modified LEnSRF algorithm outperforms the other two algorithms, with a much lower optimal λ, and an optimal ℓ similar to that of the original LEnSRF algorithm. For this model, the optimal ℓ for the LETKF algorithm is however larger than for both the original and the modified LEnSRF algorithms.

Again, the evolution of the RMSE score as a function of λ shows that the need for inflation is substantially reduced and not really needed for an ensemble of $N_\mathrm{e} = 8$ and 16 members, and even for the extreme case of an ensemble of $N_\mathrm{e} = 4$ members, which demonstrates the robustness of the modified LEnSRF algorithm.

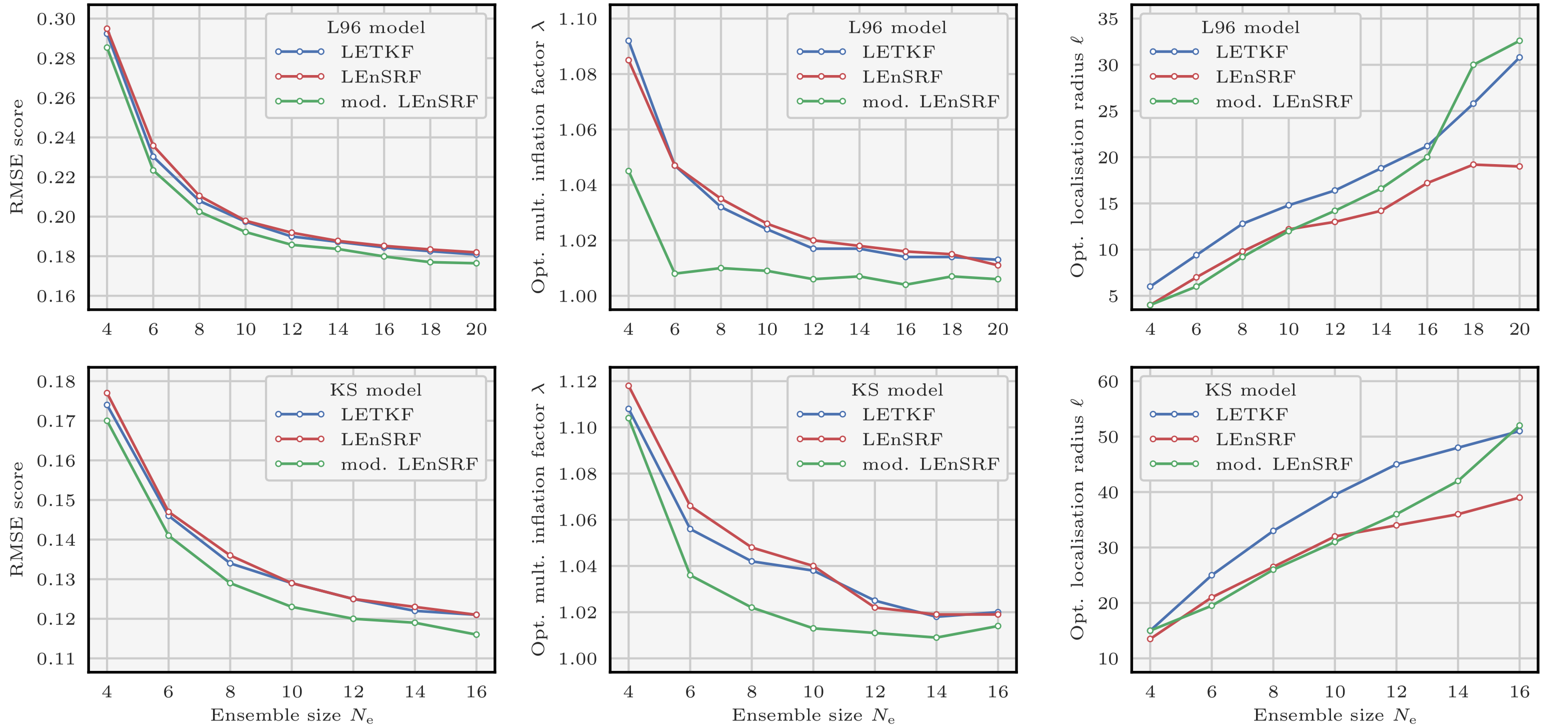

Figure 8.4: Evolution of the RMSE score (left panels), of the optimal multiplicative inflation factor λ (central panels), and of the optimal localisation radius ℓ (right panels) as a function of the ensemble size N_e for the LETKF algorithm (in blue), for the LEnSRF algorithm (in red), and for the modified LEnSRF algorithm (in green). The DA system is either the L96 model in the mildly nonlinear configuration (top panels) or the KS model in the mildly nonlinear configuration (bottom panels).

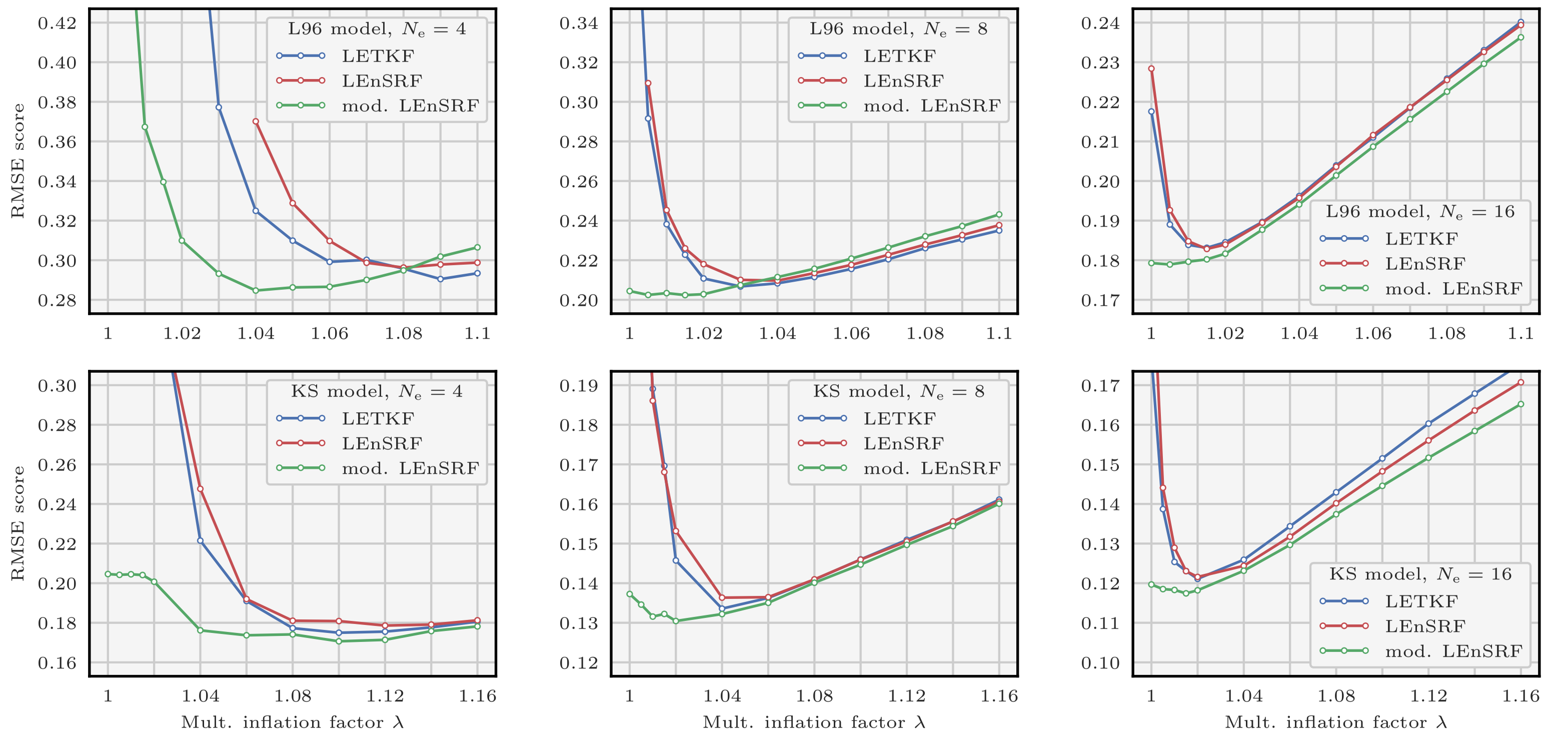

Figure 8.5: Evolution of the RMSE score as a function of multiplicative inflation factor λ for the LETKF algorithm (in blue), for the LEnSRF algorithm (in red), and for the modified LEnSRF algorithm (in green). The ensemble size is set to $N_e = 4$ (left panels), 8 (central panels), and 16 (right panels). The DA system is either the L96 model in the mildly nonlinear configuration (top panels) or the KS model in the mildly nonlinear configuration (bottom panels).

8.2.3 Robustness of the modified LEnSRF algorithm

Localisation methods can behave differently in presence of sparse and inhomogeneous observations. Moreover, we have conjectured that the new perturbation update method could generate less imbalance in the analysis perturbation matrix $\mathbf{X}^{\mathsf{a}}$. This could be evidenced with longer forecasts than those considered so far. Therefore, in this subsection, the performance of the modified LEnSRF algorithm is illustrated using twin experiments of the L96 model in two alternate configurations described in the following paragraph.

8.2.3.1 Alternate configurations for the L96 model

For the L96 model, we introduce two alternate DA configuration: the *sparse observations* and the *infrequent observations* configurations.

The sparse observations configuration is very similar to the mildly nonlinear configuration. The only difference is that the number of observations per cycle is $N_{\mathrm{y}} = d \times N_{\mathrm{x}}$, where d is the observation density, fixed in each experiment. At each assimilation cycle, the $d \times N_{\mathrm{x}}$ enabled observation sites are randomly selected (without replacement) over the total N_{x} sites. The disabled sites do not produce an observation.

The infrequent observations configuration is also very similar to the mildly nonlinear configuration. The only difference is that the time interval between consecutive observations Δt, fixed in each experiment, is longer than 0.05 unit of time. In this case, the more accurate local iterative ensemble Kalman filter would yield better RMSE scores (Bocquet 2016), but applying the new perturbation update method to this algorithm is outside the scope of this chapter.

8.2.3.2 Results

The numerical implementation for these experiments is the same as for the experiments of subsection 8.2.2. In particular, the exact same three algorithms are compared: the LETKF algorithm, the LEnSRF algorithm, and the modified LEnSRF algorithm. Figure 8.6 shows the evolution of the RMSE score and of the optimal multiplicative inflation factor λ as a function of the observation density d for the sparse observations configuration, and figure 8.7 shows the evolution of the RMSE score and of the optimal multiplicative inflation factor λ as a function of the time interval between consecutive observations Δt for the infrequent observations configuration. Furthermore, figure 8.8 shows the evolution of the RMSE score as a function of the optimal multiplicative inflation factor λ in both configurations.

For the sparse observations configuration, the results are very similar to those obtained in the experiments of subsection 8.2.2: the modified LEnSRF algorithm yields a typical 5% improvement in RMSE score, while using a significantly lower optimal λ.

For the infrequent observations configuration, again, the modified LEnSRF algorithm yields smaller RMSE score than the other two algorithms. As Δt increases, the optimal λ required to compensate for the error generated by sampling errors increases too. This is known to be due to the increased nonlinearity in the forecast step (Bocquet et al. 2015; Raanes et al. 2019a). The optimal λ required by the modified LEnSRF algorithm does increase with Δt but remains significantly smaller than the one required by the other two algorithms. By contrast with the results for the sparse observations configuration, the LETKF algorithm outperforms

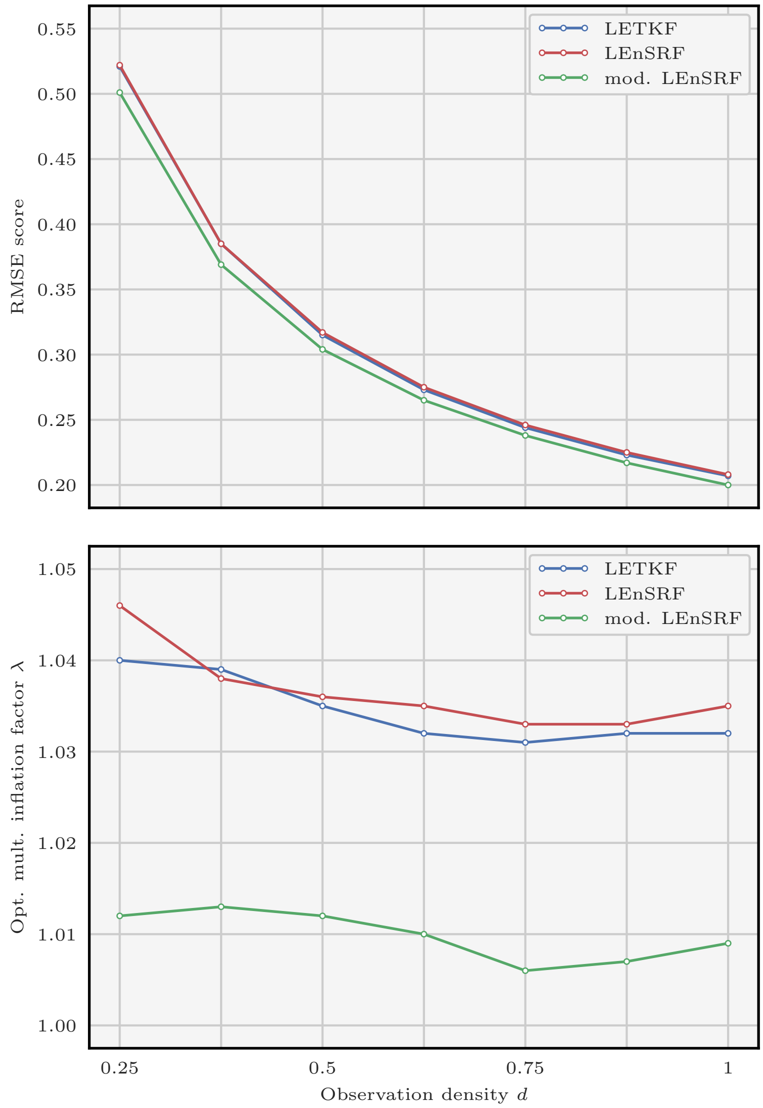

Figure 8.6: Evolution of the RMSE score (top panel) and of the optimal multiplicative inflation factor λ (bottom panel) as a function of the observation density d for the LETKF algorithm (in blue), for the LEnSRF algorithm (in red), and for the modified LEnSRF algorithm (in green). In all cases, the ensemble size N_e is set to 8 members. The DA system is the L96 model in the sparse observations configuration.

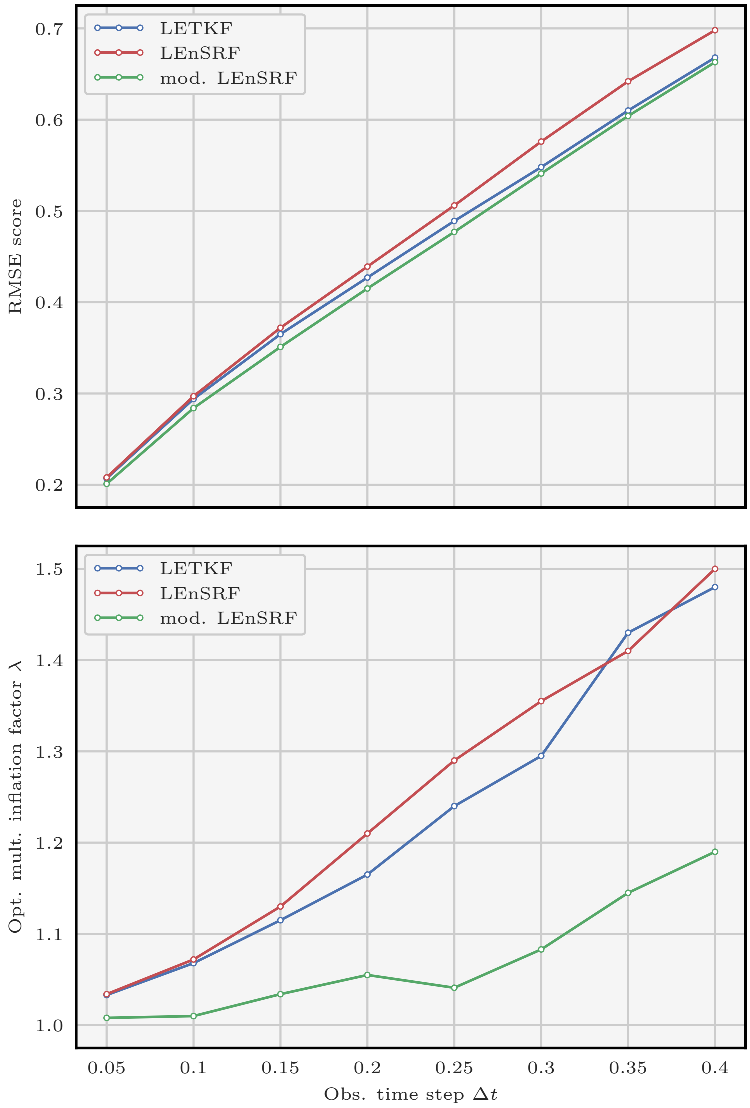

Figure 8.7: Evolution of the RMSE score (top panel) and of the optimal multiplicative inflation factor λ (bottom panel) as a function of the time interval between consecutive observations Δt for the LETKF algorithm (in blue), for the LEnSRF algorithm (in red), and for the modified LEnSRF algorithm (in green). In all cases, the ensemble size N_e is set to 8 members. The DA system is the L96 model in the infrequent observations configuration.

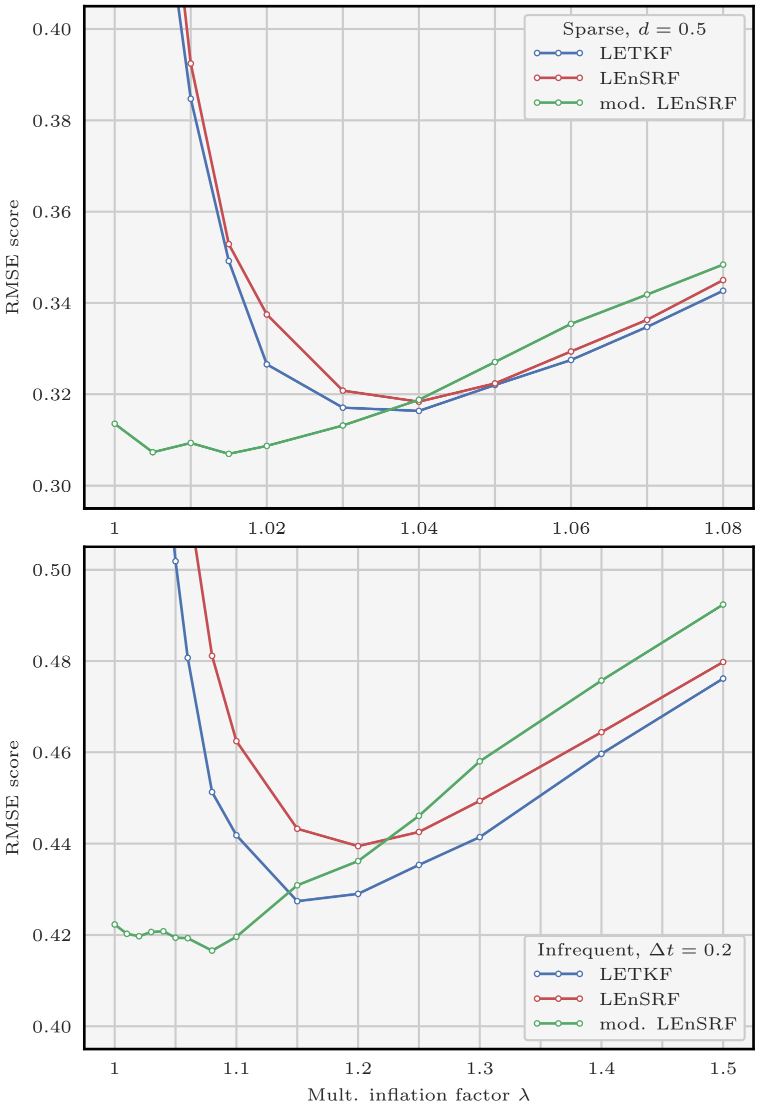

Figure 8.8: Evolution of the RMSE score as a function of the optimal multiplicative inflation factor λ for the LETKF algorithm (in blue), for the LEnSRF algorithm (in red), and for the modified LEnSRF algorithm (in green). In all cases, the ensemble size N_e is set to 8 members. The DA system is either the L96 model in the sparse observations configuration, with an observation density d set to 0.5 (top panel), or the L96 model in the infrequent observations configuration, with a time interval between consecutive observations Δt set to 0.2 unit of time (bottom panel).

the original LEnSRF algorithm and its RMSE score gets closer to that of the modified LEnSRF algorithm for large values of Δt. This supports our claim made in subsection 8.1.1 that the LETKF algorithm might generate a better forecast ensemble $\mathbf{E}^{\mathsf{f}}$.

Finally, the evolution of the RMSE score as a function of λ shows that the modified LEnSRF algorithm can yield good RMSE scores with a small λ, even in the case of sparse of infrequent observations.

In the infrequent observations configuration, we have also computed the ratio of the analysis RMSE score over the spread of the analysis ensemble $\mathbf{E}^{\mathsf{a}}$, when both the multiplicative inflation factor λ and the localisation radius ℓ are optimally tuned to yield the lowest RMSE score (not illustrated here). The modified LEnSRF and the LETKF algorithms behave quite similarly with a ratio progressively increasing from 1 to 1.10 when Δt grows from 0.05 to 0.40. By contrast, the original LEnSRF algorithm shows a ratio which increases from 1 to 1.30 in the same conditions. Again, this supports the idea that the forecast ensemble $\mathbf{E}^{\mathsf{f}}$ of the modified LEnSRF and LETKF algorithms are of better quality than that of the original LEnSRF algorithm. The same trend but progressively amplified for increasing Δt is observed for the ratio of the forecast RMSE score[4] over the spread of the forecast ensemble $\mathbf{E}^{\mathsf{f}}$.

Remark 25. These experiments have been conducted with the KS model as well (not illustrated here). The results are qualitatively very similar and yield the same conclusions for both the sparse and infrequent observations configurations.

8.2.4 Use and test of the Schatten p-norms

In this subsection, we illustrate the influence of the choice of the matrix norm in the modified LEnSRF algorithm using twin experiments of the L96 model in the mildly nonlinear configuration.

8.2.4.1 The Schatten p-norms

Let $\mathbf{M} \in \mathbb{R}^{n\times n}$ be a square matrix. The Schatten p-norm of the matrix $\mathbf{M}$ is defined as

$$\left\|\mathbf{M}\right\|_p \triangleq \left[\sum_{k=1}^{n} \sigma_k^p(\mathbf{M})\right]^{1/p}. \tag{8.37}$$

The case $p=2$ corresponds to the Frobenius norm, the case $p=1$ to the nuclear norm (the sum of the singular values), and the case $p=\infty$ to the spectral norm (the maximum of the singular values). This broad range is one strong reason why this continuum of norms is of special interest.

We now generalise the new perturbation update method to the case where the cost function $\mathcal{L}^p$ is defined by

$$\mathcal{L}^p(\mathbf{X}) \triangleq \ln\left\|\boldsymbol{\rho} \circ \left(\mathbf{X}\mathbf{X}^{\mathsf{T}}\right) - \bar{\mathbf{P}}^{\mathsf{a}}\right\|_p. \tag{8.38}$$

instead of using equation (8.8). Once again, we have chosen to apply a log-transformation to level off the ups and downs of the function. In particular, we have observed that, using

[4] The forecast RMSE score is defined, by similarity with the (analysis) RMSE score, as the RMSE between the forecast estimate $\mathbf{x}^{\mathsf{f}}$ and the truth $\mathbf{x}^{\mathsf{t}}$.

L-BFGS-B algorithm, the proposed log-transformation enables a satisfactory minimisation in the case $p = 1$ which would fail in its absence.

It is remarkable that the gradient of the cost function $\mathcal{L}^p$ can be analytically computed. Indeed, the variation of the k-th singular value of the matrix $\mathbf{M}$ is simply given by

$$\delta\sigma_k(\mathbf{M}) = \mathbf{v}_k^{\mathsf{T}}\delta\mathbf{M}\mathbf{u}_k, \tag{8.39}$$

where $\mathbf{u}_k$ and $\mathbf{v}_k$ are the normalised left and right singular vectors of $\mathbf{M}$ corresponding to the k-th singular value $\sigma_k(\mathbf{M})$. Using this relationship, we obtain the matrix gradient

$$\nabla\mathcal{L}^p(\mathbf{X}) = \frac{\nabla\|\boldsymbol{\Delta}\|_p}{\|\boldsymbol{\Delta}\|_p} = \frac{2\boldsymbol{\rho}\circ\left[\displaystyle\sum_{n=1}^{N_{\mathrm{x}}}\mathbf{u}_n\sigma_n^{p-1}(\boldsymbol{\Delta})\mathbf{v}_n^{\mathrm{T}}\right]\mathbf{X}}{\displaystyle\sum_{n=1}^{N_{\mathrm{x}}}\sigma_n^{p-1}(\boldsymbol{\Delta})}, \tag{8.40}$$

which is valid for any non-negative integer p.

Assuming that the singular values are indexed in decreasing order, in the case $p = \infty$, we have

$$\mathcal{L}^\infty(\mathbf{X}) = \ln\sigma_1(\boldsymbol{\Delta}), \tag{8.41}$$

$$\nabla\mathcal{L}^\infty(\mathbf{X}) = \frac{2\boldsymbol{\rho}\circ\left(\mathbf{u}_1\mathbf{v}_1^{\mathsf{T}}\right)\mathbf{X}}{\sigma_1(\boldsymbol{\Delta})}. \tag{8.42}$$

8.2.4.2 Accuracy of the modified LEnSRF algorithm

In this subsection, we illustrate the performance of the modified LEnSRF algorithm in which the cost function $\mathcal{L}^p$ is used in place of the cost function $\mathcal{L}$. Apart from that, the numerical implementation for these experiments is the same as for the experiments of subsection 8.2.2.

Figure 8.9 shows the evolution of the RMSE score as a function of the parameter p in the definition of the cost function $\mathcal{L}^p$. These scores are remarkably insensitive to the choice of p. However, when very close to the spectral norm limit ($p = 1$) the function minimisations seem to fail to converge (not illustrated here). We also found that the optimal multiplicative inflation factor λ and localisation radius ℓ are very similar in the whole range of p (not illustrated here). Note that, with larger values of p, the singular spectrum elevated to the p-th power is steeper and could lead to faster convergence of the minimisation.

8.3 Summary and discussion

In this chapter, we have looked back at the perturbation update in the deterministic EnKF algorithms based on CL. We have argued that the analysis perturbation matrix in the local EnKF algorithms based on CL do not represent the main modes of the analysis error covariance matrix, in contrast to the analysis perturbation matrix of the LETKF algorithm. In particular, we have focused on the LEnSRF algorithm. We have explained why equation (7.2b) still is, on theoretical grounds, a good substitute for generating the analysis perturbation matrix.

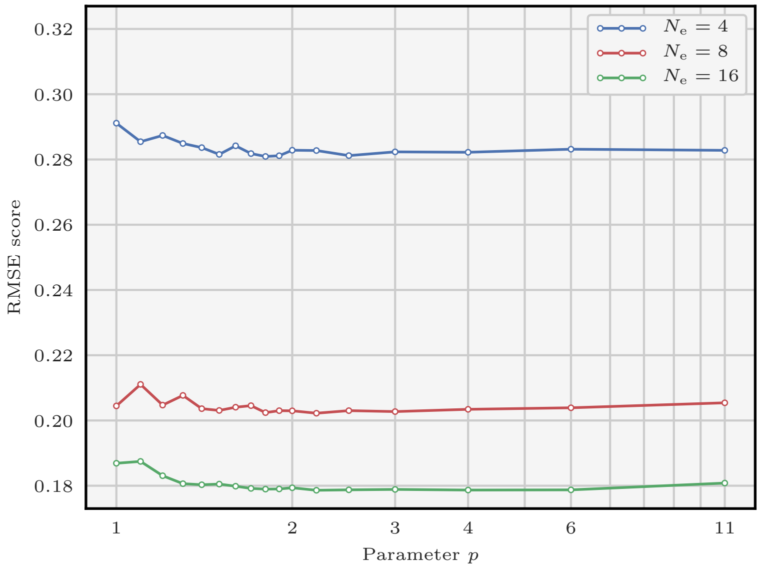

Figure 8.9: Evolution of the RMSE score as a function of the parameter p in the definition of the cost function $\mathcal{L}^p$ for the modified LEnSRF algorithm, algorithm 8.1. The ensemble size N_e is set either to 4 (in blue), to 8 (in red), or to 16 (in green). The DA system is the L96 model in the mildly nonlinear configuration.

Using these considerations, we have proposed a new perturbation update method potentially more consistent in the sense that the perturbation matrix is related to the error covariance matrix by equation (8.10) throughout the entire assimilation cycle. It consists in getting one minimiser of the cost function defined by equation (8.24). The analysis perturbation matrix is expected to be more accurate in forming short spatial separation sample covariances because less constraints are exerted on large separation sample covariances. Since we can compute the gradient of the cost function, the solution can be obtained using an off-the-shelf quasi-Newton algorithm. The evaluation of the cost function and its gradient requires a partial knowledge of the estimated analysis error covariance matrix, which is one difficulty of the method. Depending on the problem, its geometry and dimension, such knowledge could be obtained through mode expansion or through local estimations of the estimated analysis error covariance matrix.

We have tested this idea and defined a modified LEnSRF algorithm, algorithm 8.1. We have compared it numerically to the LETKF algorithm and to the original LEnSRF algorithm using two low-order one-dimensional models: the discrete 40-variable L96 model and a 128-variable spectral discretisation of the continuous Kuramoto–Sivashinsky model. We have shown that for both models, the requirement for residual multiplicative inflation still needed in spite of localisation is much weaker with the modified LEnSRF algorithm than with both the LETKF and LEnSRF algorithms. For large enough ensemble sizes, the modified LEnSRF algorithm actually performs very well without any inflation. This weaker requirement for inflation stems from a better consistency between the estimated analysis error covariance matrix and the sample covariance matrix of the analysis ensemble. We conjecture that it could be physically interpreted as a much weaker imbalance generated by the new perturbation update method. Moreover, there is an accuracy improvement of up to 6% in the RMSE score in mildly nonlinear DA configurations, which is significant in these very well tuned cases. Finally, these results have been confirmed and further strengthened in DA configurations in which the observation network is sparse or infrequent.

Conclusions

Data assimilation is the mathematical discipline which gathers all methods designed to improve the knowledge of the state of a dynamical system using both observations and modelling results of this system. In the geosciences, data assimilation it mainly applied to numerical weather prediction. It has been used in operational centres for several decades, and it has significantly contributed to the increase in quality of the forecasts.

Using ensemble methods is a powerful tool to reduce the dimension of the data assimilation systems. Currently, the two most widespread classes of ensemble data assimilation methods are the ensemble Kalman filter (EnKF) and the particle filter (PF). The success of the EnKF in high-dimensional geophysical systems is largely due to the use of localisation. Localisation is based on the assumption that correlations between state variables in a dynamical system decrease at a fast rate with the distance. In the EnKF, two localisation methods have emerged: domain localisation, and covariance localisation. Domain localisation consists of a collection of local and independent ensemble updates. This leads to efficient data assimilation algorithms, for example the local ensemble transform Kalman filter. By contrast, covariance localisation consists of a single ensemble update using a localised forecast sample covariance matrix, which is in practice much less simple to implement in a deterministic context. In the PF, the implementation of localisation is a challenge, because in this context there is no trivial way of gluing locally updated ensembles together. In this thesis, we have studied recent advances in localisation methods for ensemble data assimilation algorithms. In the first part, we have have provided an overview of the filtering methods in data assimilation. The second part has been dedicated to the implementation of localisation in the PF, and the third part to the implementation of covariance localisation in the EnKF.

In part II, we have first recalled the main results related to the weight degeneracy in the PF. In particular, we have seen that the importance sampling method suffers from the curse of dimensionality, meaning that the computational cost increases exponentially with the size of the system. Localisation can be used to counteract the curse of dimensionality. However, implementing localisation in the PF raises two major difficulties: how to glue together locally updated particles, and how to avoid imbalance in the updated ensemble. We have proposed a theoretical classification of local PF algorithms into two categories. For each category, we have presented the challenges of local particle filtering and have reviewed the ideas leading to practical implementation of the algorithms. Some of them, already in the literature, have been detailed and sometimes generalised, while others are new in this field and yield improvements in the design of the algorithms.

In the first class of algorithms, the analysis is localised by allowing the importance weight vector to vary over the grid points. The global analysis ensemble is obtained by assembling the locally updated particles, and its quality directly depends on the regularity of the local update method. This latter point is related to potential unphysical discontinuities, and hence imbalance in the assembled particles. We have presented practical methods to improve the local updates by reducing the unphysical discontinuities, the most promising being built upon the optimal transport theory. In the second class of algorithms, observations are assimilated sequentially, and localisation is introduced more generally in the analysis density by the means of a partition. The goal of the partition is to build a framework for local particle filtering without the discontinuity issue inherent to the first class of algorithms. We have shown how two methods can be used as an implementation of this framework.

The local PF algorithms have been implemented and systematically tested using twin experiments of low-order models: the Lorenz 1996 model with 40 variables and a two-dimensional model based on the barotropic vorticity equation with 1024 variables. In both models, the implementation of localisation is simple and works as expected. The local PF algorithms yield acceptable performance scores, even with small ensembles, in regimes where global the PF is degenerate. In all tested cases, using optimal transport for the local updates yields significantly better performance scores, which we have interpreted as a manifestation of mitigated unphysical discontinuities in the updated ensemble. Furthermore, the best local PF algorithms show better performance scores than the reference EnKF algorithm in a mildly nonlinear configuration of the Lorenz 1996 model. The algorithms have then been implemented in a high-resolution configuration of the barotropic vorticity model with 65 536 variables. The results confirm the conclusions of the low-order model test series, and show that the local PF algorithms may be ready to be applied to realistic geophysical systems.

Finally, we have considered the case of the prediction of tropospheric ozone concentration in western Europe during the summer 2009. Measurements of ozone concentration are available every hour at several hundreds of stations. For theses experiment, we have chosen to use the `Polair3DChemistry` model from the `Polyphemus` framework. The model has been debiased using a simple parametrisation for the bias, and the resulting debiased reference simulation yield verification scores of the same order as typical models in atmospheric chemistry. We have explained how to implement data assimilation in this system. The results show that data assimilation is effective in this system, with an effective improvement in the validation scores. Our implementation of the local PF algorithms show verification scores very similar to those of the reference EnKF algorithm. Furthermore, all ensemble data assimilation algorithms seems to have the edge over an algorithm based on pure optimal interpolation (without ensemble). Yet, it is not clear whether the small gain in validation score is sufficient to justify the huge increase in forecast wall-clock time due to the use of an ensemble.

In part III, we have first explored possible implementations for covariance localisation in deterministic EnKF algorithms using an augmented ensemble in the analysis step. We have discussed the two main difficulties with this approach: how to construct the augmented ensemble and how to update the perturbations. Two different methods have been presented to construct the augmented ensemble. The first one is based on a factorisation property and is already widespread in the geophysical data assimilation literature. As an alternative, we have proposed a second method based on randomised singular value decompositions techniques, which are very efficient when the localisation matrix is easy to apply. In both cases, the perturbation update is performed using a simple formula using linear algebra in the augmented ensemble space. The methods have been tested and compared using twin experiments of the low-order Lorenz 1996 model with 400 variables. In this case, we have found that for a given level of performance score, the second method, based on randomised techniques, requires a smaller augmented ensemble size and is hence faster than the first method.

The local EnKF algorithm with augmented ensemble has then been generalised to assimilate satellite observations in spatially extended models. In this case, covariance localisation is used in the vertical direction, while domain localisation is used in the horizontal direction. This generalised algorithm has been implemented and tested using twin experiments of a multilayer extension of the Lorenz-1996 model with a total of 1280 state variables and using a satellite-like observation operator. As expected in this system with non-local observations, the generalised algorithm yields much better performance scores than the reference EnKF algorithm, in which only domain localisation is used.

Then, we have studied the consistency of the perturbation update in deterministic EnKF algorithms using covariance localisation. We have argued that in this case, the analysis perturbations do not represent the main modes of the analysis error covariance matrix, in contrast to the analysis perturbations of EnKF algorithms using domain localisation. From these considerations, we have proposed a new perturbation update method potentially more consistent, which consists in solving an optimisation problem. The resulting analysis perturbations are expected to be more accurate in forming short spatial separation sample covariances because less constraints are exerted on large separation sample covariances. Since we can compute the gradient of the cost function, the minimisation problem can be solved using iterative minimisation algorithms. The evaluation of the cost function and its gradient requires a partial knowledge of the analysis error covariance matrix, which is one difficulty of the method.

The new perturbation update method has been tested and compared to reference local EnKF algorithms using twin experiments of two low-order models: the Lorenz-1996 model with 40 variables and a spectral discretisation of the continuous Kuramoto–Sivashinsky model with 128 variables. For both models, we have shown that the requirement for residual multiplicative inflation is much weaker with the new algorithm. Moreover, when the ensemble size is large enough, the new algorithm actually performs very well without any inflation. This weaker requirement for inflation stems from a better consistency between the analysis error covariance matrix and the sample covariance matrix of the analysis ensemble. We conjecture that it could be physically interpreted as a much weaker imbalance generated by the new algorithm. Furthermore, using the new perturbation method yield a significant improvement in the performance scores. These results have been confirmed and further strengthened in configurations in which the observation network is sparse or infrequent.

Introducing localisation in the PF is a relatively young topic and it could benefit from more theoretical and practical developments. The local update method is the main ingredient in the success, or failure, of any local PF algorithm. The approaches based on the optimal transport theory offer an elegant and efficient tool to perform the local updates while minimising the imbalance. Other approaches could be used while keeping in mind the same goals, for example the (non-optimal) transport step computed with the variational Stein descent method. In this case, the localisation could be introduced in the kernel used to compute the functional gradient of the Kullback–Leibler divergence. From a theoretical point of view, another promising approach could be to implement localisation in PF algorithms using a non-standard proposal density. In this case, it is unclear whether (and how) the proposal importance weight vector should be localised.

From a practical point of view, we have seen that the successful application of local particle filtering is largely due to the use of some kind of regularisation. In the twin experiments, the regularisation has been added under the form of a post-regularisation step, while in the atmospheric chemistry experiments, the regularisation has been added under the form of an additional model error. At this point it is clear that the design of the regularisation methods could be improved. Ideally, the regularisation should be adaptive and built concurrently with the localisation method.

The localisation frameworks introduced in this theses for the PF can only work with local observations. In the EnKF, several approximations have been introduced to combine domain localisation and non-local observations. However, non-local observations can only be rigorously assimilated when using covariance localisation instead of domain localisation. In the PF, further theoretical studies are needed to find a rigorous local assimilation method for non-local observations.

Our experiments in atmospheric chemistry demonstrate that local PF algorithms can be used with a realistic geophysical models. in this case however, using ensemble data assimilation methods did not yield significant improvement in the verification scores compared to the simpler and faster algorithm based on optimal interpolation (without an ensemble). Complementary experiments have shown that the performance of the ensemble data assimilation algorithms can be further improved by using input data perturbations. Further work is need to fix the design of the input data perturbations. Moreover, when using ensemble data assimilation, interspecies covariances are constructed during the forecast step. Therefore, another approach to improve the performance of the ensemble data assimilation could be to implement multi-species assimilation. In this case, two difficulties immediately emerge. First, we would have to precisely control the uncertainty of all assimilated species, which is delicate from a practical point of view. Second, the validation score would have to mix information between all assimilated species, and hence it would be non-trivial to define. Finally, the (moderate) success of our experiments is largely due to the use of a significant debiasing method. The debiasing is necessary here to counteract the bias in the input data, but also missing parts in the physics of the dynamical model (*e.g.*, the aerosols). Data assimilation experiments with a full model (including aerosols) would probably require a much less important debiasing while yielding better verification scores.

The use of covariance localisation in the EnKF becomes increasingly important with the prominence of satellite observations. The new perturbation update method offer an elegant implementation of covariance localisation in the EnKF, but it could still benefit from more theoretical and practical development. This method has been tested in numerical experiments with low-order models, in which computing the exact analysis error covariance matrix is possible. In realistic applications, the augmented ensemble approach could be used to obtain an approximation of the analysis error covariance matrix. In this case, it would be desirable to check that the advantages of the new method – reduced need for multiplicative inflation and increased accuracy – remain valid. This reduced need for inflation stems from a better consistency between the estimated analysis error covariance matrix and the sample covariance matrix of the analysis ensemble. We conjecture that it could be physically interpreted as a much weaker imbalance generated by the new perturbation update method. This hypothesis could be validated using numerical experiments of complex two- or three-dimensional models, in which balance between different physical variables are important. Finally, following the augmented ensemble approach, the new method could also be generalised to assimilate satellite radiances in a spatially extended models. In this case, the generalised algorithm could be tested, for example, using twin simulations of the multilayer Lorenz-1996 model.

Bibliography

Acevedo, Walter, Jana de Wiljes and Sebastian Reich (2017). 'Second-order Accurate Ensemble Transform Particle Filters'. In: *SIAM Journal on Scientific Computing* 39.5, A1834–A1850 (cit. on p. 63).

Ades, Melanie and Peter Jan van Leeuwen (2013). 'An exploration of the equivalent weights particle filter'. In: *Quarterly Journal of the Royal Meteorological Society* 139.672, pp. 820–840 (cit. on pp. 68, 80).

— (2015). 'The equivalent-weights particle filter in a high-dimensional system'. In: *Quarterly Journal of the Royal Meteorological Society* 141.687, pp. 484–503 (cit. on p. 120).

Anderson, Jeffrey L. (1996). 'A Method for Producing and Evaluating Probabilistic Forecasts from Ensemble Model Integrations'. In: *Journal of Climate* 9.7, pp. 1518–1530 (cit. on p. 116).

Anderson, Jeffrey L. and Lili Lei (2013). 'Empirical Localization of Observation Impact in Ensemble Kalman Filters'. In: *Monthly Weather Review* 141.11, pp. 4140–4153 (cit. on pp. 43, 212).

Arbogast, Étienne, Gérald Desroziers and Loïk Berre (2017). 'A parallel implementation of a 4DEnVar ensemble'. In: *Quarterly Journal of the Royal Meteorological Society* 143.706, pp. 2073–2083 (cit. on p. 196).

Arulampalam, M. Sanjeev, Simon Maskell, Neil Gordon and Tim Clapp (2002). 'A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking'. In: *IEEE Transactions on Signal Processing* 50.2, pp. 174–188 (cit. on p. 46).

Asch, Mark, Marc Bocquet and Maëlle Nodet (2016). *Data assimilation: methods, algorithms, and applications*. Fundamentals of algorithms 11. Philadelphia: SIAM, Society for Industrial and Applied Mathematics. 306 pp. (cit. on pp. 5, 28, 36).

Balgovind, Ramesh, Amnon Dalcher, Michael Ghil and Eugenia Kalnay (1983). 'A Stochastic-Dynamic Model for the Spatial Structure of Forecast Error Statistics'. In: *Monthly Weather Review* 111.4, pp. 701–722 (cit. on p. 168).

Bengtsson, Thomas, Chris Snyder and Doug Nychka (2003). 'Toward a nonlinear ensemble filter for high-dimensional systems'. In: *Journal of Geophysical Research: Atmospheres* 108.D24 (cit. on p. 98).

Bessagnet, Bertrand et al. (2016). 'Presentation of the EURODELTA III intercomparison exercise – evaluation of the chemistry transport models' performance on criteria pollutants and joint analysis with meteorology'. In: *Atmospheric Chemistry and Physics* 16.19, pp. 12667–12701 (cit. on p. 164).

Bishop, Craig H., Brian J. Etherton and Sharanya J. Majumdar (2001). 'Adaptive Sampling with the Ensemble Transform Kalman Filter. Part I: Theoretical Aspects'. In: *Monthly Weather Review* 129.3, pp. 420–436 (cit. on pp. 28, 63).

Bishop, Craig H. and Daniel Hodyss (2009). 'Ensemble covariances adaptively localized with ECO-RAP. Part 2: a strategy for the atmosphere'. In: *Tellus A* 61.1, pp. 97–111 (cit. on p. 194).

— (2011). 'Adaptive Ensemble Covariance Localization in Ensemble 4D-VAR State Estimation'. In: *Monthly Weather Review* 139.4, pp. 1241–1255 (cit. on p. 194).

Bishop, Craig H., Jeffrey S. Whitaker and Lili Lei (2017). 'Gain Form of the Ensemble Transform Kalman Filter and Its Relevance to Satellite Data Assimilation with Model Space Ensemble Covariance Localization'. In: *Monthly Weather Review* 145.11, pp. 4575–4592 (cit. on pp. 194, 198, 200, 212, 219).

Bocquet, Marc (2011). 'Ensemble Kalman filtering without the intrinsic need for inflation'. In: *Nonlinear Processes in Geophysics* 18.5, pp. 735–750 (cit. on p. 36).

— (2014). *Introduction to the principles and methods of data assimilation in the geosciences* (cit. on p. 5).

— (2016). 'Localization and the iterative ensemble Kalman smoother'. In: *Quarterly Journal of the Royal Meteorological Society* 142.695, pp. 1075–1089 (cit. on pp. 194, 198–199, 218, 227–228, 239).

Bocquet, Marc and Alberto Carrassi (2017). 'Four-dimensional ensemble variational data assimilation and the unstable subspace'. In: *Tellus A: Dynamic Meteorology and Oceanography* 69.1, p. 1304504 (cit. on p. 43).

Bocquet, Marc and Alban Farchi (2019). 'On the consistency of the local ensemble square root Kalman filter perturbation update'. In: *Tellus A: Dynamic Meteorology and Oceanography* 71.1, pp. 1–21 (cit. on pp. 198–199, 221).

Bocquet, Marc, Karthik S. Gurumoorthy, Amit Apte, Alberto Carrassi, Colin Grudzien and Christopher K. R. T. Jones (2017). 'Degenerate Kalman Filter Error Covariances and Their Convergence onto the Unstable Subspace'. In: *SIAM/ASA Journal on Uncertainty Quantification* 5.1, pp. 304–333 (cit. on p. 209).

Bocquet, Marc, Carlos A. Pires and Lin Wu (2010). 'Beyond Gaussian Statistical Modeling in Geophysical Data Assimilation'. In: *Monthly Weather Review* 138.8, pp. 2997–3023 (cit. on pp. 71, 75, 78, 80, 82).

Bocquet, Marc, Patrick N. Raanes and Alexis Hannart (2015). 'Expanding the validity of the ensemble Kalman filter without the intrinsic need for inflation'. In: *Nonlinear Processes in Geophysics* 22.6, pp. 645–662 (cit. on pp. 95, 239).

Bocquet, Marc and Pavel Sakov (2013). 'Joint state and parameter estimation with an iterative ensemble Kalman smoother'. In: *Nonlinear Processes in Geophysics* 20.5, pp. 803–818 (cit. on p. 36).

Boynard, Anne, Matthias Beekmann, Gilles Foret, Anthony Ung, Sophie Szopa, Catherine Schmechtig and Adriana Coman (2011). 'An ensemble assessment of regional ozone model uncertainty with an explicit error representation'. In: *Atmospheric Environment* 45.3, pp. 784–793 (cit. on p. 186).

Brankart, Jean-Michel, Emmanuel Cosme, Charles-Emmanuel Testut, Pierre Brasseur and Jacques Verron (2011). 'Efficient Local Error Parameterizations for Square Root or Ensemble Kalman Filters: Application to a Basin-Scale Ocean Turbulent Flow'. In: *Monthly Weather Review* 139.2, pp. 474–493 (cit. on p. 194).

Brasseur, Pierre and Jacques Verron (2006). 'The SEEK filter method for data assimilation in oceanography: a synthesis'. In: *Ocean Dynamics* 56.5-6, pp. 650–661 (cit. on p. 21).

Browne, Philip A. (2016). 'A comparison of the equivalent weights particle filter and the local ensemble transform Kalman filter in application to the barotropic vorticity equation'. In: *Tellus A: Dynamic Meteorology and Oceanography* 68.1, p. 30466 (cit. on p. 120).

Buehner, Mark (2005). 'Ensemble-derived stationary and flow-dependent background-error covariances: Evaluation in a quasi-operational NWP setting'. In: *Quarterly Journal of the Royal Meteorological Society* 131.607, pp. 1013–1043 (cit. on p. 194).

Bunch, Pete and Simon Godsill (2016). 'Approximations of the Optimal Importance Density Using Gaussian Particle Flow Importance Sampling'. In: *Journal of the American Statistical Association* 111.514, pp. 748–762 (cit. on p. 66).

Burgers, Gerrit, Peter Jan van Leeuwen and Geir Evensen (1998). 'Analysis Scheme in the Ensemble Kalman Filter'. In: *Monthly Weather Review* 126.6, pp. 1719–1724 (cit. on p. 22).

Byrd, Richard H., Peihuang Lu, Jorge Nocedal and Ciyou Zhu (1995). 'A Limited Memory Algorithm for Bound Constrained Optimization'. In: *SIAM Journal on Scientific Computing* 16.5, pp. 1190–1208 (cit. on pp. 163, 226).

Campbell, William F., Craig H. Bishop and Daniel Hodyss (2010). 'Vertical Covariance Localization for Satellite Radiances in Ensemble Kalman Filters'. In: *Monthly Weather Review* 138.1, pp. 282–290 (cit. on pp. 194, 211–212).

Carrassi, Alberto, Marc Bocquet, Laurent Bertino and Geir Evensen (2018). 'Data assimilation in the geosciences: An overview of methods, issues, and perspectives'. In: *Wiley Interdisciplinary Reviews: Climate Change* 9.5, e535 (cit. on pp. 5, 38, 41).

Cheng, Yuan and Sebastian Reich (2015). 'Assimilating data into scientific models: An optimal coupling perspective'. In: van Leeuwen, Peter Jan, Yuan Cheng and Sebastian Reich. *Nonlinear Data Assimilation.* Vol. 2. Cham: Springer International Publishing, pp. 75–118 (cit. on pp. 82, 93, 107).

Chopin, Nicolas (2004). 'Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference'. In: *The Annals of Statistics* 32.6, pp. 2385–2411 (cit. on p. 69).

Chorin, Alexandre J., Matthias Morzfeld and Xuemin Tu (2010). 'Implicit particle filters for data assimilation'. In: *Communications in Applied Mathematics and Computational Science* 5.2, pp. 221–240 (cit. on p. 68).

Chorin, Alexandre J. and Xuemin Tu (2009). 'Implicit sampling for particle filters'. In: *Proceedings of the National Academy of Sciences* 106.41, pp. 17249–17254 (cit. on p. 68).

Chustagulprom, Nawinda, Sebastian Reich and Maria Reinhardt (2016). 'A Hybrid Ensemble Transform Particle Filter for Nonlinear and Spatially Extended Dynamical Systems'. In: *SIAM/ASA Journal on Uncertainty Quantification* 4.1, pp. 592–608 (cit. on p. 134).

Cohn, Stephen E. (1997). 'An Introduction to Estimation Theory'. In: *Journal of the Meteorological Society of Japan. Ser. II* 75 (1B), pp. 257–288 (cit. on p. 5).

Crisan, Dan and Arnaud Doucet (2002). 'A survey of convergence results on particle filtering methods for practitioners'. In: *IEEE Transactions on Signal Processing* 50.3, pp. 736–746 (cit. on pp. 69–70).

Del Moral, Pierre (1996). 'Non Linear Filtering: Interacting Particle Solution'. In: *Markov Processes and Related Fields* 2, pp. 555–580 (cit. on p. 69).

— (1999). 'Central Limit Theorem for Nonlinear Filtering and Interacting Particle Systems'. In: *The Annals of Applied Probability* 9.2, pp. 275–297 (cit. on p. 69).

Del Moral, Pierre and Laurent Miclo (2000). ‘Branching and interacting particle systems. Approximations of Feynman-Kac formulae with applications to non-linear filtering’. In: *Séminaire de probabilités de Strasbourg* 34. Ed. by Springer – Lecture Notes in Mathematics, pp. 1–145 (cit. on p. 69).

Desroziers, Gérald, Etienne Arbogast and Loïk Berre (2016). ‘Improving spatial localization in 4DEnVar: Improving Spatial Localization in 4DEnVar’. In: *Quarterly Journal of the Royal Meteorological Society* 142.701, pp. 3171–3185 (cit. on pp. 196, 228).

Desroziers, Gérald, Jean-Thomas Camino and Loïk Berre (2014). ‘4DEnVar: link with 4D state formulation of variational assimilation and different possible implementations: 4DEnVar: State Formulation of 4D-Var and Possible Implementations’. In: *Quarterly Journal of the Royal Meteorological Society* 140.684, pp. 2097–2110 (cit. on p. 196).

Dezső, Balázs, Alpár Jüttner and Péter Kovács (2011). ‘LEMON – an Open Source C++ Graph Template Library’. In: *Electronic Notes in Theoretical Computer Science* 264.5, pp. 23–45 (cit. on p. 139).

Douc, Randal and Eric Moulines (2008). ‘Limit theorems for weighted samples with applications to sequential Monte Carlo methods’. In: *The Annals of Statistics* 36.5, pp. 2344–2376 (cit. on p. 69).

Doucet, Arnaud, Nando Freitas and Neil Gordon, eds. (2001). *Sequential Monte Carlo Methods in Practice*. New York, NY: Springer New York (cit. on pp. 45, 53).

Doucet, Arnaud, Simon Godsill and Christophe Andrieu (2000). ‘On sequential Monte Carlo sampling methods for Bayesian filtering’. In: *Statistics and Computing* 10.3, pp. 197–208 (cit. on pp. 67–68).

Doucet, Arnaud and Adam M. Johansen (2011). ‘Tutorial on Particle Filtering and Smoothing: Fifteen Years Later’. In: *The Oxford Handbook of Nonlinear Filtering*. Ed. by Dan Crisan and Boris Rozovskii. Oxford University Press, pp. 656–704 (cit. on pp. 46, 49, 60).

Eckart, Carl and Gale Young (1936). ‘The approximation of one matrix by another of lower rank’. In: *Psychometrika* 1.3, pp. 211–218 (cit. on p. 206).

Evensen, Geir (1994). ‘Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics’. In: *Journal of Geophysical Research* 99.C5, p. 10143 (cit. on p. 22).

Farchi, Alban and Marc Bocquet (2018). ‘Review article: Comparison of local particle filters and new implementations’. In: *Nonlinear Processes in Geophysics* 25.4, pp. 765–807 (cit. on pp. 76, 86, 115).

— (2019). ‘On the efficiency of covariance localisation of the ensemble Kalman filter using augmented ensembles’. In: *Frontiers in Applied Mathematics and Statistics* 5 (cit. on p. 194).

Fertig, Elana J., Brian R. Hunt, Edward Ott and Istvan Szunyogh (2007). ‘Assimilating non-local observations with a local ensemble Kalman filter’. In: *Tellus A: Dynamic Meteorology and Oceanography* 59.5, pp. 719–730 (cit. on pp. 193, 211).

Gaspari, Gregory and Stephen E. Cohn (1999). ‘Construction of correlation functions in two and three dimensions’. In: *Quarterly Journal of the Royal Meteorological Society* 125.554, pp. 723–757 (cit. on p. 40).

Gaubert, Benjamin (2013). ‘Assimilation des observations pour la modélisation de la qualité de l’air’. PhD thesis. Paris: Université Paris-VII (cit. on pp. 166, 169).

Geweke, John (1989). ‘Bayesian Inference in Econometric Models Using Monte Carlo Integration’. In: *Econometrica* 57.6, p. 1317 (cit. on p. 49).

Golub, Gene H. and Charles F. Van Loan (2013). *Matrix computations*. Fourth edition. Johns Hopkins studies in the mathematical sciences. OCLC: ocn824733531. Baltimore: The Johns Hopkins University Press. 756 pp. (cit. on p. 226).

Gordon, Neil J., David J. Salmond and A.F.M. Smith (1993). ‘Novel approach to nonlinear/non-Gaussian Bayesian state estimation’. In: *IEE Proceedings F Radar and Signal Processing* 140.2, p. 107 (cit. on pp. 45, 56).

Greybush, Steven J., Eugenia Kalnay, Takemasa Miyoshi, Kayo Ide and Brian R. Hunt (2011). ‘Balance and Ensemble Kalman Filter Localization Techniques’. In: *Monthly Weather Review* 139.2, pp. 511–522 (cit. on p. 43).

Guenther, Alex, Thomas Karl, Peter Harley, Christine Wiedinmyer, Paul I. Palmer and Chris Geron (2006). ‘Estimates of global terrestrial isoprene emissions using MEGAN (Model of Emissions of Gases and Aerosols from Nature)’. In: *Atmospheric Chemistry and Physics* 6.11, pp. 3181–3210 (cit. on p. 161).

Halko, N., P. G. Martinsson and J. A. Tropp (2011). ‘Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions’. In: *SIAM Review* 53.2, pp. 217–288 (cit. on p. 202).

Hamill, Thomas M. and Stephen J. Colucci (1997). ‘Verification of Eta–RSM Short-Range Ensemble Forecasts’. In: *Monthly Weather Review* 125.6, pp. 1312–1327 (cit. on p. 116).

Hamill, Thomas M., Jeffrey S. Whitaker and Chris Snyder (2001). ‘Distance-Dependent Filtering of Background Error Covariance Estimates in an Ensemble Kalman Filter’. In: *Monthly Weather Review* 129.11, pp. 2776–2790 (cit. on pp. 39, 193).

Haussaire, Jean-Matthieu (2017). ‘Méthodes variationnelles d’ensemble itératives pour l’assimilation de données non-linéaire : Application au transport et la chimie atmosphérique’. PhD thesis. Champs-sur-Marne: Université Paris-Est (cit. on pp. 157, 167, 169, 174).

Horn, Roger A. and Charles R. Johnson (2012). *Matrix analysis*. Second edition, corrected reprint. New York, NY: Cambridge University Press. 643 pp. (cit. on pp. 28, 39, 198).

Horowitz, Larry W. et al. (2003). ‘A global simulation of tropospheric ozone and related tracers: Description and evaluation of MOZART, version 2’. In: *Journal of Geophysical Research: Atmospheres* 108.D24, n/a–n/a (cit. on p. 161).

Houtekamer, Peter L. and Herschel L. Mitchell (1998). ‘Data Assimilation Using an Ensemble Kalman Filter Technique’. In: *Monthly Weather Review* 126.3, pp. 796–811 (cit. on p. 22).

— (2001). ‘A Sequential Ensemble Kalman Filter for Atmospheric Data Assimilation’. In: *Monthly Weather Review* 129.1, pp. 123–137 (cit. on pp. 41, 193).

Hunt, Brian R., Eric J. Kostelich and Istvan Szunyogh (2007). ‘Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter’. In: *Physica D: Nonlinear Phenomena* 230.1-2, pp. 112–126 (cit. on pp. 28, 36, 41).

Kalman, Rudolf E. (1960). ‘A New Approach to Linear Filtering and Prediction Problems’. In: *Journal of Basic Engineering* 82.1, p. 35 (cit. on p. 21).

Kalnay, Eugenia, Yoichiro Ota, Takemasa Miyoshi and Junjie Liu (2012). ‘A simpler formulation of forecast sensitivity to observations: application to ensemble Kalman filters’. In: *Tellus A: Dynamic Meteorology and Oceanography* 64.1, p. 18462 (cit. on p. 228).

Kassam, Aly-Khan and Lloyd N. Trefethen (2005). 'Fourth-Order Time-Stepping for Stiff PDEs'. In: *SIAM Journal on Scientific Computing* 26.4, pp. 1214–1233 (cit. on pp. 198, 235).

Kepert, Jeffrey D. (2009). 'Covariance localisation and balance in an Ensemble Kalman Filter'. In: *Quarterly Journal of the Royal Meteorological Society* 135.642, pp. 1157–1176 (cit. on p. 43).

Kitagawa, Genshiro (1996). 'Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models'. In: *Journal of Computational and Graphical Statistics* 5.1, p. 1 (cit. on p. 60).

Kong, Augustine, Jun S. Liu and Wing Hung Wong (1994). 'Sequential Imputations and Bayesian Missing Data Problems'. In: *Journal of the American Statistical Association* 89.425, p. 278 (cit. on pp. 49, 54, 61–62, 67, 75).

Kretschmer, Matthew, Brian R. Hunt and Edward Ott (2015). 'Data assimilation using a climatologically augmented local ensemble transform Kalman filter'. In: *Tellus A: Dynamic Meteorology and Oceanography* 67.1, p. 26617 (cit. on p. 194).

Künsch, Hans R. (2005). 'Recursive Monte Carlo filters: Algorithms and theoretical analysis'. In: *The Annals of Statistics* 33.5, pp. 1983–2021 (cit. on pp. 60, 69).

Kuramoto, Yoshiki and Toshio Tsuzuki (1975). 'On the Formation of Dissipative Structures in Reaction-Diffusion Systems: Reductive Perturbation Approach'. In: *Progress of Theoretical Physics* 54.3, pp. 687–699 (cit. on p. 234).

— (1976). 'Persistent Propagation of Concentration Waves in Dissipative Media Far from Thermal Equilibrium'. In: *Progress of Theoretical Physics* 55.2, pp. 356–369 (cit. on p. 234).

Law, Kody, Andrew Stuart and Konstantinos Zygalakis (2015). *Data Assimilation.* Vol. 62. Texts in Applied Mathematics. Cham: Springer International Publishing (cit. on pp. 5, 11).

Le Gland, François, Valérie Monbet and Vu-Duc Tran (2011). 'Large sample asymptotics for the ensemble Kalman filter'. In: *The Oxford Handbook of Nonlinear Filtering.* Ed. by Dan Crisan and Boris Rozovskii. Oxford University Press, pp. 598–631 (cit. on p. 33).

Lee, Yoonsang and Andrew J. Majda (2016). 'State estimation and prediction using clustered particle filters'. In: *Proceedings of the National Academy of Sciences* 113.51, pp. 14609–14614 (cit. on pp. 82, 85).

Leng, Hongze, Junqiang Song, Fengshun Lu and Xiaoqun Cao (2013). 'A New Data Assimilation Scheme: The Space-Expanded Ensemble Localization Kalman Filter'. In: *Advances in Meteorology* 2013, pp. 1–6 (cit. on p. 194).

Liu, Qiang and Dilin Wang (2016). 'Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm'. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems.* NIPS'16. Barcelona, Spain: Curran Associates Inc., pp. 2378–2386 (cit. on p. 65).

Lorenc, Andrew C. (2003). 'The potential of the ensemble Kalman filter for NWP—a comparison with 4D-Var'. In: *Quarterly Journal of the Royal Meteorological Society* 129.595, pp. 3183–3203 (cit. on pp. 194, 200).

— (2017). 'Improving ensemble covariances in hybrid variational data assimilation without increasing ensemble size'. In: *Quarterly Journal of the Royal Meteorological Society* 143.703, pp. 1062–1072 (cit. on p. 194).

Lorenz, Edward N. and Kerry A. Emanuel (1998). ‘Optimal Sites for Supplementary Weather Observations: Simulation with a Small Model’. In: *Journal of the Atmospheric Sciences* 55.3, pp. 399–414 (cit. on p. 117).

Louis, Jean-François (1979). ‘A parametric model of vertical eddy fluxes in the atmosphere’. In: *Boundary-Layer Meteorology* 17.2, pp. 187–202 (cit. on p. 161).

MacKay, David J. C. (2003). *Information theory, inference, and learning algorithms.* Cambridge, UK ; New York: Cambridge University Press. 628 pp. (cit. on p. 81).

Mallet, Vivien et al. (2007). ‘Technical Note: The air quality modeling system Polyphemus’. In: *Atmospheric Chemistry and Physics* 7.20, pp. 5479–5487 (cit. on pp. 157, 160).

Mandel, Jan, Loren Cobb and Jonathan D. Beezley (2011). ‘On the convergence of the ensemble Kalman filter’. In: *Applications of Mathematics* 56.6, pp. 533–541 (cit. on p. 33).

Manton, Jonathan H., Robert Mahony and Yingbo Hua (2003). ‘The geometry of weighted low-rank approximations’. In: *IEEE Transactions on Signal Processing* 51.2, pp. 500–514 (cit. on p. 224).

Ménétrier, Benjamin, Thibaut Montmerle, Yann Michel and Loïk Berre (2015a). ‘Linear Filtering of Sample Covariances for Ensemble-Based Data Assimilation. Part I: Optimality Criteria and Application to Variance Filtering and Covariance Localization’. In: *Monthly Weather Review* 143.5, pp. 1622–1643 (cit. on p. 43).

— (2015b). ‘Linear Filtering of Sample Covariances for Ensemble-Based Data Assimilation. Part II: Application to a Convective-Scale NWP Model’. In: *Monthly Weather Review* 143.5, pp. 1644–1664 (cit. on p. 43).

Metref, Sammy, Emmanuel Cosme, Chris Snyder and Pierre Brasseur (2014). ‘A non-Gaussian analysis scheme using rank histograms for ensemble data assimilation’. In: *Nonlinear Processes in Geophysics* 21.4, pp. 869–885 (cit. on p. 100).

Miyoshi, Takemasa and Yoshiaki Sato (2007). ‘Assimilating Satellite Radiances with a Local Ensemble Transform Kalman Filter (LETKF) Applied to the JMA Global Model (GSM)’. In: *SOLA* 3, pp. 37–40 (cit. on pp. 193, 212).

Morzfeld, Matthias, Daniel Hodyss and Chris Snyder (2017). ‘What the collapse of the ensemble Kalman filter tells us about particle filters’. In: *Tellus A: Dynamic Meteorology and Oceanography* 69.1, p. 1283809 (cit. on p. 69).

Morzfeld, Matthias, Xuemin Tu, Ethan Atkins and Alexandre J. Chorin (2012). ‘A random map implementation of implicit filters’. In: *Journal of Computational Physics* 231.4, pp. 2049–2066 (cit. on p. 68).

Musso, Christian, Nadia Oudjane and François Le Gland (2001). ‘Improving Regularised Particle Filters’. In: *Sequential Monte Carlo Methods in Practice.* Ed. by Arnaud Doucet, Nando Freitas and Neil Gordon. New York, NY: Springer New York, pp. 247–271 (cit. on pp. 62, 95, 123, 126).

Ott, Edward, Brian R. Hunt, Istvan Szunyogh, Aleksey V. Zimin, Eric J. Kostelich, Matteo Corazza, Eugenia Kalnay, D. J. Patil and James A. Yorke (2004). ‘A local ensemble Kalman filter for atmospheric data assimilation’. In: *Tellus A* 56.5, pp. 415–428 (cit. on pp. 32, 41, 193).

Papadakis, Nicolas, Etienne Mémin, Anne Cuzol and Nicolas Gengembre (2010). ‘Data assimilation with the weighted ensemble Kalman filter’. In: *Tellus A: Dynamic Meteorology and Oceanography* 62.5, pp. 673–697 (cit. on p. 69).

Pele, Ofir and Michael Werman (2009). 'Fast and robust Earth Mover's Distances'. In: *2009 IEEE 12th International Conference on Computer Vision.* 2009 IEEE 12th International Conference on Computer Vision (ICCV). Kyoto: IEEE, pp. 460–467 (cit. on p. 109).

Penny, Stephen G., David W. Behringer, James A. Carton and Eugenia Kalnay (2015). 'A Hybrid Global Ocean Data Assimilation System at NCEP'. In: *Monthly Weather Review* 143.11, pp. 4660–4677 (cit. on p. 212).

Penny, Stephen G. and Takemasa Miyoshi (2016). 'A local particle filter for high-dimensional geophysical systems'. In: *Nonlinear Processes in Geophysics* 23.6, pp. 391–405 (cit. on pp. 82, 84–85, 89–90, 107, 135).

Peres-Neto, Pedro R., Donald A. Jackson and Keith M. Somers (2005). 'How many principal components? stopping rules for determining the number of non-trivial axes revisited'. In: *Computational Statistics & Data Analysis* 49.4, pp. 974–997 (cit. on p. 200).

Pham, Dinh Tuan, Jacques Verron and Marie Christine Roubaud (1998). 'A singular evolutive extended Kalman filter for data assimilation in oceanography'. In: *Journal of Marine Systems* 16.3-4, pp. 323–340 (cit. on p. 21).

Pitt, Michael K. and Neil Shephard (1999). 'Filtering via Simulation: Auxiliary Particle Filters'. In: *Journal of the American Statistical Association* 94.446, p. 590 (cit. on p. 68).

Poterjoy, Jonathan (2016). 'A Localized Particle Filter for High-Dimensional Nonlinear Systems'. In: *Monthly Weather Review* 144.1, pp. 59–76 (cit. on pp. 83, 86, 95–96, 102–104, 107, 118, 143).

Poterjoy, Jonathan, Louis Wicker and Mark Buehner (2019). 'Progress toward the Application of a Localized Particle Filter for Numerical Weather Prediction'. In: *Monthly Weather Review* 147.4, pp. 1107–1126 (cit. on p. 86).

Pulido, Manuel and Peter Jan van Leeuwen (2019). 'Sequential Monte Carlo with kernel embedded mappings: The mapping particle filter'. In: *Journal of Computational Physics* (cit. on p. 65).

Raanes, Patrick N., Marc Bocquet and Alberto Carrassi (2019a). 'Adaptive covariance inflation in the ensemble Kalman filter by Gaussian scale mixtures'. In: *Quarterly Journal of the Royal Meteorological Society* 145.718, pp. 53–75 (cit. on pp. 38–39, 239).

Raanes, Patrick N., Alberto Carrassi and Laurent Bertino (2015). 'Extending the Square Root Method to Account for Additive Forecast Noise in Ensemble Methods'. In: *Monthly Weather Review* 143.10, pp. 3857–3873 (cit. on pp. 25–26, 31).

Raanes, Patrick N., Andreas S. Stordal and Geir Evensen (2019b). 'Revising the stochastic iterative ensemble smoother'. In: *Nonlinear Processes in Geophysics Discussions.* In review, pp. 1–22 (cit. on p. 35).

Rebeschini, Patrick and Ramon van Handel (2015). 'Can local particle filters beat the curse of dimensionality?' In: *The Annals of Applied Probability* 25.5, pp. 2809–2866 (cit. on pp. 79, 82, 84–85, 87, 107).

Reich, Sebastian (2013). 'A Nonparametric Ensemble Transform Method for Bayesian Inference'. In: *SIAM Journal on Scientific Computing* 35.4, A2013–A2024 (cit. on pp. 63–64, 107, 134).

Reich, Sebastian and Colin Cotter (2015). *Probabilistic Forecasting and Bayesian Data Assimilation.* Cambridge: Cambridge University Press (cit. on p. 63).

Ripley, Brian D., ed. (1987). *Stochastic Simulation.* Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc. (cit. on p. 59).

Robert, Christian P. and George Casella (2004). *Monte Carlo Statistical Methods.* Springer Texts in Statistics. New York, NY: Springer New York (cit. on p. 60).

Robert, Sylvain and Hans R. Künsch (2017). 'Localizing the Ensemble Kalman Particle Filter'. In: *Tellus A: Dynamic Meteorology and Oceanography* 69.1, p. 1282016 (cit. on pp. 93, 97, 104–105, 107, 111).

Rubin, Donald B., ed. (1987). *Multiple Imputation for Nonresponse in Surveys.* Wiley Series in Probability and Statistics. Hoboken, NJ, USA: John Wiley & Sons, Inc. (cit. on pp. 45, 56).

Sakov, Pavel and Laurent Bertino (2011). 'Relation between two common localisation methods for the EnKF'. In: *Computational Geosciences* 15.2, pp. 225–237 (cit. on p. 44).

Sakov, Pavel and Peter R. Oke (2008a). 'A deterministic formulation of the ensemble Kalman filter: an alternative to ensemble square root filters'. In: *Tellus A: Dynamic Meteorology and Oceanography* 60.2, pp. 361–371 (cit. on p. 195).

— (2008b). 'Implications of the Form of the Ensemble Transformation in the Ensemble Square Root Filters'. In: *Monthly Weather Review* 136.3, pp. 1042–1053 (cit. on p. 32).

Sakov, Pavel, Dean S. Oliver and Laurent Bertino (2012). 'An Iterative EnKF for Strongly Nonlinear Systems'. In: *Monthly Weather Review* 140.6, pp. 1988–2004 (cit. on p. 36).

Shen, Zheqi, Youmin Tang and Xiaojing Li (2017). 'A new formulation of vector weights in localized particle filters'. In: *Quarterly Journal of the Royal Meteorological Society* 143.709, pp. 3269–3278 (cit. on p. 86).

Silverman, B. W. (1986). *Density estimation for statistics and data analysis.* 1., CRC Press repr. Monographs on statistics and applied probability 26. OCLC: 248204797. Boca Raton, Fla.: Chapman & Hall/CRC. 175 pp. (cit. on pp. 62, 75–76, 95).

Sivashinsky, Gregory I. (1977). 'Nonlinear analysis of hydrodynamic instability in laminar flames—I. Derivation of basic equations'. In: *Acta Astronautica* 4.11-12, pp. 1177–1206 (cit. on p. 234).

Snyder, Chris (2014). 'Introduction to the Kalman filter'. In: *Advanced Data Assimilation for Geosciences.* Ed. by Éric Blayo, Marc Bocquet, Emmanuel Cosme and Leticia F. Cugliandolo. Oxford University Press, pp. 75–120 (cit. on p. 38).

Snyder, Chris, Thomas Bengtsson, Peter Bickel and Jeff Anderson (2008). 'Obstacles to High-Dimensional Particle Filtering'. In: *Monthly Weather Review* 136.12, pp. 4629–4640 (cit. on pp. 46, 75, 77–79, 82).

Snyder, Chris, Thomas Bengtsson and Mathias Morzfeld (2015). 'Performance Bounds for Particle Filters Using the Optimal Proposal'. In: *Monthly Weather Review* 143.11, pp. 4750–4761 (cit. on pp. 68, 80–81).

Spantini, Alessio, Daniele Bigoni and Youssef Marzouk (2018). 'Inference via Low-Dimensional Couplings'. In: *Journal of Machine Learning Research* 19.66, pp. 1–71 (cit. on p. 66).

Srebro, Nathan and Tommi Jaakkola (2003). 'Weighted Low-rank Approximations'. In: *Proceedings of the Twentieth International Conference on International Conference on Machine Learning.* Twentieth International Conference on International Conference on Machine Learning. Ed. by Tom Fawcett and Nina Mishra. Washington, DC, USA: AAAI Press, pp. 720–727 (cit. on p. 224).

Talagrand, Olivier, Robert Vautard and Bernard Strauss (1997). 'Evaluation of probabilistic prediction systems'. In: *Workshop on Predictability, 20-22 October 1997.* Workshop on

Predictability, 20-22 October 1997. Shinfield Park, Reading: ECMWF, pp. 1–26 (cit. on p. 116).

Van Leeuwen, Peter Jan (2003). 'A Variance-Minimizing Filter for Large-Scale Applications'. In: *Monthly Weather Review* 131.9, pp. 2071–2084 (cit. on pp. 71, 75).

— (2009). 'Particle Filtering in Geophysical Systems'. In: *Monthly Weather Review* 137.12, pp. 4089–4114 (cit. on pp. 46, 71, 75–76, 80, 82, 107).

— (2010). 'Nonlinear data assimilation in geosciences: an extremely efficient particle filter'. In: *Quarterly Journal of the Royal Meteorological Society* 136.653, pp. 1991–1999 (cit. on pp. 68, 80).

Van Leeuwen, Peter Jan and Melanie Ades (2013). 'Efficient fully nonlinear data assimilation for geophysical fluid dynamics'. In: *Computers & Geosciences* 55, pp. 16–27 (cit. on p. 120).

Van Leeuwen, Peter Jan, Hans R. Künsch, Lars Nerger, Roland Potthast and Sebastian Reich (2019). 'Particle filters for high-dimensional geoscience applications: A review'. In: *Quarterly Journal of the Royal Meteorological Society* (cit. on p. 46).

Vestreng, Vigdis, Justin Goodwin and Martin M. Adams (2004). *Inventory Review 2004. Emission data reported to CLRTAP and under the NEC Directive.* Technical report MSW 1/2004. EMEP/EEA (cit. on p. 161).

Villani, Cédric (2009). *Optimal Transport.* Vol. 338. Grundlehren der mathematischen Wissenschaften. Berlin, Heidelberg: Springer Berlin Heidelberg (cit. on p. 64).

Whitaker, Jeffrey S. and Thomas M. Hamill (2002). 'Ensemble Data Assimilation without Perturbed Observations'. In: *Monthly Weather Review* 130.7, pp. 1913–1924 (cit. on p. 196).

Wu, Lin, Vivien Mallet, Marc Bocquet and Bruno Sportisse (2008). 'A comparison study of data assimilation algorithms for ozone forecasts'. In: *Journal of Geophysical Research* 113.D20 (cit. on p. 186).

Yarwood, Greg, Sunja Rao, Rowland Way, Mark Yocke, Gary Z Whitten and Smog Reyes (2005). 'Updates to the Carbon Bond Chemical Mechanism: CB05'. In: 9th Annual CMAS Conference. Chapel Hill, NC, p. 246 (cit. on p. 160).

Zhang, Leiming, Jeffrey R. Brook and Robert Vet (2003). 'A revised parameterization for gaseous dry deposition in air-quality models'. In: *Atmospheric Chemistry and Physics* 3.6, pp. 2067–2082 (cit. on p. 161).

Zhou, Yuhua, Dennis McLaughlin and Dara Entekhabi (2006). 'Assessing the Performance of the Ensemble Kalman Filter for Land Surface Data Assimilation'. In: *Monthly Weather Review* 134.8, pp. 2128–2142 (cit. on pp. 71, 75).

Zhu, Mengbin, Peter Jan van Leeuwen and Javier Amezcua (2016). 'Implicit equal-weights particle filter'. In: *Quarterly Journal of the Royal Meteorological Society* 142.698, pp. 1904–1919 (cit. on p. 68).

Zupanski, Milija (2005). 'Maximum Likelihood Ensemble Filter: Theoretical Aspects'. In: *Monthly Weather Review* 133.6, pp. 1710–1726 (cit. on p. 36).

Acronyms

3D–Var	three-dimensional variational analysis 20–23, 26, 39
4D–Var	four-dimensional variational analysis 22
APF	auxiliary particle filter 73
BLUE	best linear unbiased estimate 18–23, 26, 38, 170
BV	barotropic vorticity 124, 126–128, 154–162
cdf	cumulative density function 99, 144, 176
CL	covariance localisation 43–45, 47, 48, 102, 109, 200, 201, 210, 216–218, 222–227, 233, 246
CTM	chemistry and transport model 163, 164, 166, 167, 170, 187, 195
DA	data assimilation 9–16, 18–23, 25, 26, 28, 40, 41, 43, 47, 49, 50, 53, 54, 58, 59, 69, 72–75, 79–86, 118, 121–123, 126–132, 134–140, 142, 143, 145, 147–149, 151, 152, 154–165, 170, 172–174, 176–192, 194–196, 200, 202, 210, 212, 215, 216, 222–224, 226, 238–244, 247
DEnKF	deterministic ensemble Kalman filter 200, 201, 204
DL	domain localisation 45, 47, 48, 86, 89, 90, 200, 216–218, 222, 224
ECMWF	European centre for medium-range weather forecasts 167
EKF	extended Kalman filter 38
EMEP	European monitoring and evaluation programme 167
EnKF	ensemble Kalman filter 5, 6, 25–39, 41, 43–45, 47, 49, 50, 59, 67, 73, 79, 86, 89, 90, 99, 102, 109, 128, 129, 143, 150, 153, 155, 156, 162, 189, 196, 199, 200, 216, 223, 225, 226, 238, 246, 249–253
EnKPF	ensemble Kalman particle filter 97, 108, 112
ETKF	ensemble transform Kalman filter 27, 32, 34, 36–40, 42, 45, 128, 129, 131, 150, 151, 153–155, 157, 158, 202, 204
ETPF	ensemble transform particle filter 68, 69, 97, 111, 117, 132, 155
EWPF	equivalent-weights particle filter 73, 84
GC	Gaspari–Cohn 43, 44, 47, 86, 90, 95, 175, 211, 219
GL	additive Gaussian and linear 16–18, 20–22, 25–27, 29, 32–34, 46, 72, 201, 205, 228
GLCF	global land cover facility 167
HMM	hidden Markov model 10–13
iid	idenpendant and identically distributed 1, 4, 28, 33, 37, 39, 41, 50, 52, 64, 66, 67, 69, 81, 83, 133

IS	importance sampling 52–54, 57, 58, 61, 64, 66, 75, 80–84, 86, 93
KDDM	kernel density distribution mapping 100, 108
KDE	kernel density estimation 67, 99, 111
KF	Kalman filter 25–34, 37, 38, 72
KS	Kuramoto–Sivashinsky 237–241, 245
L^{2}EnSRF	local analysis local ensemble square root Kalman filter 218, 219, 222–224
L96	Lorenz 1996 123–126, 128–134, 136–140, 142, 143, 145, 147–154, 156, 161, 214–216, 218, 219, 222, 224, 226, 237–245, 247
LEnSRF	local ensemble square root Kalman filter 200, 201, 204, 205, 214–217, 223–230, 237–247
LET	linear ensemble transform 67, 68, 71, 97, 114, 143
LETKF	local ensemble transform Kalman filter 45–47, 90, 128, 129, 131, 132, 134, 137, 145, 147–149, 151, 153–156, 159–161, 172, 173, 175, 176, 178–180, 185, 187, 190–195, 199, 215–217, 219, 221–224, 226, 227, 230, 233, 238–247
LPF	local particle filter 80, 86–90, 94, 95, 100, 101, 106, 108, 111, 118, 119, 121, 124, 131, 133, 136, 140, 148, 149, 153, 156, 161–163, 173, 177, 187, 195
LPF–X	first class of LPF algorithms 87–89, 91–95, 97–99, 101, 108, 111, 112, 114, 117, 118, 133, 135–148, 150, 151, 153–157, 159–161, 172, 176, 180–185, 187, 190–195
LPF–Y	second class of LPF algorithms 101–103, 105, 108–111, 114–116, 118, 146–151, 153–156, 158, 160, 161, 172, 176, 180, 185, 187, 195
MAE	mean absolute error 177, 185–187, 190–192, 194
MC	Monte Carlo 41, 49–53, 66, 81
MEGAN	model of emissions of gases and aerosols from nature 167
mL96	multilayer Lorenz 1996 218–220, 223, 224
MLEF	maximum likelihood ensemble filter 40
MOZART	model for ozone and related chemical tracers 167
MRHF	multivariate rank histogram filter 104, 106
NWP	numerical weather prediction 9, 22
ODE	ordinary differential equation 123, 124, 218, 219
OI	optimal interpolation 172–178, 180, 181, 185, 192, 195
PDE	partial differential equation 237
pdf	probability density function 1, 4, 10, 12, 13, 16, 51, 54, 64, 67, 69, 72, 74, 80, 88, 93, 99, 100, 104, 122
PF	particle filter 5, 6, 49, 50, 59–63, 65, 67, 70, 71, 73–75, 79, 80, 84, 86–89, 91, 93, 104, 107, 108, 111, 117–119, 128, 129, 131, 132, 153, 156, 161, 162, 189, 196, 249–252
RMSE	root-mean-square error 122, 129–134, 136–151, 153–161, 168, 170, 177–180, 185, 187, 190–193, 195, 214–216, 222–224, 238–247
SA	single analysis 17, 18, 20, 21

SEEK	singular evolutive extended Kalman 25
SIR	sampling importance resampling 49, 60, 61, 68, 71, 73–75, 84, 87, 117, 118, 130–134, 136, 155
SIS	sequential importance sampling 49, 50, 57–59, 61, 73
svd	singular value decomposition 207–216, 221–224, 227
VMPF	variational mapping particle filter 70
WEnKF	weighted ensemble Kalman filter 73
WLRA	weighted low-rank approximation 227, 228

Abstract

Data assimilation is the mathematical discipline which gathers all the methods designed to improve the knowledge of the state of a dynamical system using both observations and modelling results of this system. In the geosciences, data assimilation it mainly applied to numerical weather prediction. It has been used in operational centres for several decades, and it has significantly contributed to the increase in quality of the forecasts.

Ensemble methods are powerful tools to reduce the dimension of the data assimilation systems. Currently, the two most widespread classes of ensemble data assimilation methods are the ensemble Kalman filter (EnKF) and the particle filter (PF). The success of the EnKF in high-dimensional geophysical systems is largely due to the use of localisation. Localisation is based on the assumption that correlations between state variables in a dynamical system decrease at a fast rate with the distance. In this thesis, we have studied and improved localisation methods for ensemble data assimilation.

The first part is dedicated to the implementation of localisation in the PF. The recent developments in local particle filtering are reviewed, and a generic and theoretical classification of local PF algorithms is introduced, with an emphasis on the advantages and drawbacks of each category. Alongside the classification, practical solutions to the difficulties of local particle filtering are suggested. The local PF algorithms are tested and compared using twin experiments with low- to medium-order systems. Finally, we consider the case study of the prediction of the tropospheric ozone using concentration measurements. Several data assimilation algorithms, including local PF algorithms, are applied to this problem and their performances are compared.

The second part is dedicated to the implementation of covariance localisation in the EnKF. We show how covariance localisation can be efficiently implemented in the deterministic EnKF using an augmented ensemble. The proposed algorithm is tested using twin experiments with a medium-order model and satellite-like observations. Finally, the consistency of the deterministic EnKF with covariance localisation is studied in details. A new implementation is proposed and compared to the original one using twin experiments with low-order models.

Keywords: data assimilation, particle filter, ensemble Kalman filter,
localisation, geosciences, atmospheric chemistry

Résumé

L'assimilation de données est la discipline permettant de combiner des observations d'un système dynamique avec un modèle numérique simulant ce système, l'objectif étant d'améliorer la connaissance de l'état du système. Le principal domaine d'application de l'assimilation de données est la prévision numérique du temps. Les techniques d'assimilation sont implémentées dans les centres opérationnels depuis plusieurs décennies et elles ont largement contribué à améliorer la qualité des prédictions.

Une manière efficace de réduire la dimension des systèmes d'assimilation de données est d'utiliser des méthodes ensemblistes. La plupart de ces méthodes peuvent être regroupées en deux classes : le filtre de Kalman d'ensemble (EnKF) et le filtre particulaire (PF). Le succès de l'EnKF pour des problèmes géophysiques de grande dimension est largement dû à la localisation. La localisation repose sur l'hypothèse que les corrélations entre variables d'un système dynamique décroissent très rapidement avec la distance. Dans cette thèse, nous avons étudié et amélioré les méthodes de localisation pour l'assimilation de données ensembliste.

La première partie est dédiée à l'implémentation de la localisation dans le PF. Nous passons en revue les récents développements concernant la localisation dans le PF et nous proposons une classification théorique des algorithmes de type PF local. Nous insistons sur les avantages et les inconvénients de chaque catégorie puis nous proposons des solutions pratiques aux problèmes que posent les PF localisés. Les PF locaux sont testés et comparés en utilisant des expériences jumelles avec des modèles de petite et moyenne dimension. Finalement, nous considérons le cas de la prédiction de l'ozone troposphérique en utilisant des mesures de concentration. Plusieurs algorithmes, dont des PF locaux, sont implémentés et appliqués à ce problème et leurs performances sont comparées.

La deuxième partie est dédiée à l'implémentation de la localisation des covariances dans l'EnKF. Nous montrons comment la localisation des covariances peut être efficacement implémentée dans l'EnKF déterministe en utilisant un ensemble augmenté. L'algorithme obtenu est testé au moyen d'expériences jumelles avec un modèle de moyenne dimension et des observations satellitaires. Finalement, nous étudions en détail la cohérence de l'EnKF déterministe avec localisation des covariances. Une nouvelle méthode est proposée puis comparée à la méthode traditionnelle en utilisant des simulation jumelles avec des modèles de petite dimension.

Mots-clés : assimilation de données, filtre particulaire, filtre de Kalman d'ensemble,
localisation, geosciences, chimie atmosphérique