UNIVERSITÀ DEGLI STUDI DI TRIESTE

XXXI CICLO DEL DOTTORATO DI RICERCA IN
SCIENZE DELLA TERRA E MECCANICA DEI FLUIDI

BIOGEOCHEMICAL DATA ASSIMILATION: ON THE SINGULAR EVOLUTIVE INTERPOLATED KALMAN-FILTER

Settore scientifico-disciplinare: **MAT/08**

DOTTORANDO
SIMONE SPADA

COORDINATORE
PROF. PIERPAOLO OMARI

SUPERVISORE DI TESI
DR. STEFANO SALON

CO-SUPERVISORI DI TESI
DR. GIANPIERO COSSARINI
PROF. STEFANO MASET

ANNO ACCADEMICO 2017/2018

Contents

List of Figures

List of Tables

Abstract

Data Assimilation is nowadays a fundamental part in any forecasting geoscience model.

In the field of marine biogeochemistry, the number and quality of observational systems (e.g. satellites, bio argo floats, moorings) is constantly improving, but the available information is still far from picturing *per se* the true state of the marine ecosystem.

Thus, at forecasting and monitoring purpose, Data Assimilation techniques are necessary in order to face the problem of the ocean state estimation.

Handling the large dimension of the state vector of the system (order of 10^6) remains an issue, and many attempts have been done in literature to reduce the complexity of the problem, adding hypotheses and approximations in order to obtain fast Data Assimilation algorithms. The 3D-VAR, a Data Assimilation method based on the variational approach, is one of these results and is adopted in the EU Copernicus forecast system MedBFM, which is responsible for monitoring and forecasting the biogeochemical state of the Mediterranean Sea.

Aside from variational, the other main Data Assimilation approach is the Kalman-Filter.

This thesis is focused on biogeochemical marine Data Assimilation, and has a double purpose.

The first one is to compare the 3D-VAR scheme with the Singular Evolutive Interpolated Kalman-Filter (SEIK).

From a theoretical point of view, this is realized using a Bayesian framework to derive differences and similarities as well as strengths and weaknesses of the two methods. This analysis shows that the main differences are in the choice of the state estimator (the mode for the variational and the mean for the Kalman-Filter), and in the Kalman-Filter's capability of keeping and transferring the information through the time steps.

A twin experiment has been used to assess the skill performance of the compared schemes. Tests show that the SEIK is one order of magnitude more precise than the 3D-VAR, in terms of root mean squared distance (RMSD).

The second objective of this work is to develop a novel Data Assimilation method from the SEIK filter, focusing on the effects of the model error and its estimation. Various strategies have been implemented at this purpose, namely a high order sampling technique, a method to take into account SEIK's neglected part of the model error as noise-like observation error, a data-driven maximum likelihood algorithm for model error estimation and,

finally, a computationally cheap *ad hoc* smoother.

The twin experiment tests prove that the first two modifications change the behavior of the SEIK filter only in case of large model error, conferring to the modified SEIK a higher resiliency to divergences. The maximum likelihood strategy estimations obtained good agreement with the estimated real value, with better results if used in pair with the modified version of the SEIK. The smoother further improved the RMSD of the method, with even better results in case of large model error.

Chapter 1

Introduction

In biogeochemical marine modelling, the estimation of the current state of the system is a key element in order to successfully simulate the future dynamics. Unfortunately, the available data (from satellites and other measurement equipments) often offer only an incomplete information by which is not easy (or impossible) to effortlessly derive a sufficiently good approximation of the state. The term Data Assimilation (DA) encompasses all those techniques used to extrapolate, from available data and model, the best estimation of the present (analysis), past (reanalysis) and future (forecast) states.

Geosciences suffer the very big dimension of the state vector of the system (order of 10^6) and handling this difficulty remains an issue, also considering the computational efficiency required by operational systems to satisfy performance and provide short-term forecast products to users. Indeed, the majority of the operations (e.g. products, inversions etc.) involving the covariance matrices are too heavy to be computed for operational forecasting purposes, also considering the increasing horizontal resolution. Various attempts have been made in literature to reduce the complexity of these tasks, often adding hypotheses to simplify the problem and decrease the computational cost (e.g. [11]). However, computing power availability has strongly increased recently, and some of the constraints used in the past can be now partially revised.

Focusing on marine biogeochemistry and taking as an example the EU Copernicus forecast system MedBFM ([27]), which is responsible for monitoring and forecasting the biogeochemical state of the Mediterranean Sea, assimilates satellite chlorophyll data through a 3D-VAR method ([44]). The 3D-VAR is an algorithm based on the variational approach, one of the most common techniques to manage DA in operational context.

The first objective of this work is to compare the 3D-VAR with the Singular Evolutive Interpolated Kalman-Filter algorithm (SEIK, [37]), where the

latter is a method based on a Kalman-Filter, which is the other mainstream approach to Data Assimilation.
At this purpose, the thesis presents (Chapters 2, 3 and 4) a theoretical part that uses a Bayesian framework to derive similarities and differences between the two approaches, and points out some advantages and disadvantages.
A twin-experiment is used to test the skill of the two algorithms on a biogeochemical model coupled with a physical advection-diffusion transport model (Chapter 6). The experiment is based on a Fasham-like model simulated in a 2D square domain representing the photic zone. Even if not complex as the cited operational biogeochemical system, the experiment produce complex dynamics that allows to test the properties of the different DA methods.
The second objective of this work is to develop and test a novel Data Assimilation method from the SEIK algorithm, focusing in particular on effects and estimation of the model uncertainties. Chapter 5 contains all the strategies and modifications made to the SEIK, namely a high order sampling method, a noise-like interpretation of the unconsidered model error and a maximum likelihood approach to uncertainty estimation. Chapter 6 includes and discusses the experiment done to test the proposed methods.
Chapter 7 derives and tests an *ad hoc* smoother.
Chapter 8 presents the implementation of the novel Data Assimilation scheme in a realistic 3D model, namely the OGSTM-BFM (the computational core of the MedBFM model system), in order to assess the feasibility of the method and show an efficient parallelization method. Finally, the conclusive chapter summarizes the obtained results and proposes some future work ideas.

List of original contributions of this thesis:

- The SEIK filter has been generalised including weights in the sampling.
- A new sampling method has been developed, with a higher order of convergence in the most relevant PCA components of the error subspace.
- A new method to fully take into account model error has been developed.
- A maximum likelihood strategy to model error estimation has been presented.
- An *ad hoc* smoother has been developed
- An efficient parallelization method for the novel filter implementation has been presented.

Chapter 2

Data Assimilation and Variational approach

After a briefly presentation of the Data Assimilation problem, this Chapter is focused on the variational approach solution.
Each variational method is based on the minimization of some error functional.
From a computational point of view, the cheapest (and fastest) scheme is the 3D-VAR.
4D-VAR instead is a more complex alternative, but its functional is based on the assumption of a model without errors.

2.1 What is Data Assimilation?

Data Assimilation main purpose is the estimation of the state of a dynamical system (e.g. the concentrations of biogeochemical tracers into the sea), taking into account the simulations computed by a numerical model and observations data. Both simulations and data suffer a certain uncertainty, the former due to, for example, numerical errors, sub-grid processes, unknown boundaries and initial conditions or unmodelled events, while the latter caused by incomplete information, experimental and representativeness (or representation) errors.
Data Assimilation can try to estimate a state in the present (analysis), past (reanalysis) and therefore help with future state estimations (forecast). In prediction-oriented systems, present and future states have a predominant role, while reanalysis is more interesting for scientific and monitoring purpose.
In order to enter into details of the formalisation of the Data Assimilation

state estimation problem, it is important to spend a few moment to set up notations.

2.2 Notations

This thesis follows (with some minor differences) the notations introduced in [21].
The state vector of the system at time t_i is indicated with $x_i \in \mathcal{X}_i \cong \mathbb{R}^N$. The dimension N of the state vector space $\mathcal{X}_i$ takes into account all the degrees of freedom of the system. If for example 3 concentration variables (say phytoplankton, zooplankton and nutrients) are modelled inside a cubic domain of $10 \times 10 \times 10$ grid points, then $N = 3 \times 10 \times 10 \times 10 = 3000$.
The index $i \in \{0, \ldots, K\}$, with K a positive integer, enumerate the times (in chronological order, i.e. $t_i < t_j$ for $i < j$) at which an estimation of the system is desired. Colons between indices, as in the writing $x_{i:j}$, with $i < j$, is a contraction to say $(x_i, \ldots, x_j) \in \mathcal{X}_i \times \cdots \times \mathcal{X}_j \cong \mathbb{R}^{N(j-i+1)}$.
The dynamical system evolves from time t_{i-1} to t_i via the model operator

$$m_i : \mathcal{X}_{i-1} \to \mathcal{X}_i,$$

that represents all the computations made by the numerical model to integrate the system. Parameters, boundary conditions and forcing used by the model is implicitly included in this function. To some extent,

$$m_i\left(x_{i-1}\right) \approx x_i,$$

where the symbol $\approx$ is used here because the computed evolution only approximates the real state at the next time, and a certain uncertainty holds.
The observation datum vector measured at time t_i is indicated with $y_i \in \mathcal{Y}_i \cong \mathbb{R}^n$. Data come from measurements made by various instruments, like satellites, floats, ferrybox, moorings etc. and they are affected by errors. n is the dimension of the datum vector space and indicates the number of measurements gathered at time t_i. Using the previous example, if a satellite measures the chlorophyll at surface with a 8×8 resolution grid while a mooring gets 5 vertical measurements of all the 3 variables, then $n = 8 \times 8 + 3 \times 5 = 79$.
The observation operator

$$h_i : \mathcal{X}_i \to \mathcal{Y}_i,$$

maps the state x_i in the theoretical observation numerically computed from x_i. Obviously, due to errors and approximations, in general it is not equal to the real datum, so

$$h_i\left(x_i\right) \approx y_i.$$

Very often, as in the example above, $n \ll N$ and each datum carries only an incomplete information, in the sense that it alone is not enough to go back from y_i to the corresponding state x_i, or, mathematically speaking, h is not invertible.
Comparing real data with simulations,

$$h_i\left(m_i\left(x_{i-1}\right)\right) = y_i$$

rarely holds, due to uncertainties involved with x_{i-1}, m_i and h_i. Then, a method to estimate states and errors is needed, namely a Data Assimilation method.
In the following parts, the best estimation of the state x_i taking into account all the previous data $y_{0:i-1}$ but before receiving the datum y_i, will be indicated with x_i^f, where the f is for "forecast". The estimation after the assimilation of the datum y_i instead, is called "analysis" and indicated with x_i^a. Further, x_i^r is the "reanalysis" state, or the estimation calculated using all the available data $y_{0:K}$. The more general notion of "background" state x_{i-1}^b indicates a known estimation without specifying if it is from a forecast, an analysis or obtained in some other way.

2.3 Variational approach

To handle the state estimation problem, the variational approach relay on least squares techniques.
The strategy is to define a suitable error functional J and minimize it.
A first option is

$$J\left(x_{0:i}\right) := \left\|x_0 - x_0^b\right\|^2_{B_0^{-1}} + \sum_{i=1}^{K} \left\|x_i - m_i\left(x_{i-1}\right)\right\|^2_{Q_i^{-1}} + \sum_{i=0}^{K} \left\|h_i\left(x_i\right) - y_i\right\|^2_{R_i^{-1}}, \tag{2.1}$$

where

$$\|x\|^2_M := x^T M x$$

is the euclidean norm weighted with the symmetric positive-definite matrix M. The first addend in (2.1) represents the background error in the initial state x_0, the second one is the model error and the third is the error on the observations. Different matrices have been used to calibrate the significance of each error and how to choose them is debatable, but using covariance matrices seems a quite natural option (a probabilistic explanation will be presented in Section 3.4).
In geoscience, where the dimension N is big, this error functional is hardly

managed. A possible simplification is obtained neglecting the model error and substituting in equation (2.1) x_i with $m_i\left(x_{i-1}\right)$ for every i or, equivalently,

$$x_i = m_i \circ \cdots \circ m_1\left(x_0\right).$$

The obtained functional $J^{Q=0}$ now depends only on the initial state x_0:

$$J^{Q=0}\left(x_0\right) := \left\|x_0 - x_0^b\right\|^2_{B_0^{-1}} + \sum_{i=0}^{K} \left\|h_i \circ m_i \circ \cdots \circ m_1\left(x_0\right) - y_i\right\|^2_{R_i^{-1}}. \qquad (2.2)$$

Minimizing $J^{Q=0}$ is more affordable than working with J, and this method is quite diffused in geophysical Data Assimilation, with the name of 4D-VAR (see [7] for a complete overview).

Nevertheless, in biogeochemical sea Data Assimilation, another simplification of (2.1), namely the 3D-VAR, is often preferred ([44]): instead of the model error, the background error on the previous state x_{i-1} is neglected, i.e.

$$x_{i-1} = x_{i-1}^b,$$

and the functional J is divided in $K+1$ functionals J_i,

$$J_i\left(x_i\right) := \left\|x_i - m_i\left(x_{i-1}^b\right)\right\|^2_{Q_i^{-1}} + \left\|h_i\left(x_i\right) - y_i\right\|^2_{R_i^{-1}}, \qquad (2.3)$$

with each J_i depending only on x_i. Thus, the analysis state is

$$x_i^a = \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}}\, J_i\left(x_i\right), \qquad (2.4)$$

and it will be used as background state at the next step.

The main advantage of the 3D-VAR is its cheap computational cost. In fact, the evaluation of m_i can be expensive and, while minimizing (2.2) requires m_i many times in each iteration, working with (2.3) is much easier. Furthermore, the minimization process needs the gradient of the error functional and, in the 4D-VAR case, this implies the use of the adjoint of m_i, which can be a not trivial issue to manage.

In the last years, the vast improvement in information technologies suggests to push over the computational side, and, while the 4D-VAR seems the natural (computationally heavier) successor of the 3D-VAR, it is not perfectly suited for the biogeochemical field. In fact, the no model error assumption in (2.2) is a quite good approximation in physical contexts, but it can be debatable in biogeochemical systems: the laws (and equations) behind complex biological populations' behaviors are not as known as the physical laws and are often more probabilistic then deterministic. Thus, some uncertainty on the model should be taken into account.

Chapter 3

Bayesian theory and Kalman approach

Instead of least squares variational approach presented in the previous section, Kalman Filters are the other main method in the field of state estimation. The first "optimal filter", as Kalman called it in 1960 ([25]), has led to a ever growing variety of filters that find a broad spectrum of applications in scientific and engineering modelling (see [14] or [34] for an overview). The Unscented Filter ([24]), the Ensemble Kalman Filter ([12]) and the Particle Filter ([32]) are probably the most common examples of such evolution in filter theory.

In this chapter, a probabilistic framework is provided as a starting point to derive the Kalman-Filter scheme while preparing a common ground useful to make the comparison with the 3D-VAR method.

At this purpose, the 3D-VAR it self is derived again, showing that it is based over some of the Kalman-Filter's assumptions.

On the other hand, the differences between the two methods are shown as well. In fact, 3D-VAR uses the mode as estimator, instead of the mean. Further, the Kalman-Filter passes the covariance information from one time step to the next, while 3D-VAR neglects it.

3.1 Premises and more notations

Since this chapter is focused on probabilities and Bayesian inference, a few more definitions are needed to set up the framework. First, let's define $\mathcal{P}\left(\mathcal{X}_i\right)$ and $\mathcal{P}\left(\mathcal{Y}_i\right)$ such that

$$\mathcal{P}\left(V\right) = \left\{f : V \to \mathbb{R} \text{ such that } f \text{ is a probability density function over } V\right\}.$$

In the following, using a simple notation, the small p with no other indexes means "probability of", the vertical line inside the argument is the symbol for conditioned probability and the comma is the logical *AND* operator. So, for example, "$p\left(x_i\right)$" and "$p\left(x_i|x_{i-1},y_i\right)$" read as "probability of x_i" and "probability of x_i, knowing that x_{i-1} and y_i" respectively.
In this set up, a probabilistic model operator $\mathcal{M}_i$ is needed, in order to consider both the evolution and the related uncertainties:

$$\begin{gathered}\mathcal{M}_i : \mathcal{X}_{i-1} \to \mathcal{P}\left(\mathcal{X}_i\right),\\ \mathcal{M}_i\left(x_{i-1}\right) := p_{\mathcal{M}_i\left(x_{i-1}\right)},\\ p_{\mathcal{M}_i\left(x_{i-1}\right)}\left(x_i\right) := p\left(x_i|x_{i-1}\right).\end{gathered} \tag{3.1}$$

This definition comes from the hypothesis that, given x_{i-1}, no other previous states matter for x_i and for the possible errors. This is also commonly known as Markov property, and represents the memorylessness of the system (see [16] for more on Markov processes).
Another common and reasonable assumption is that errors on measurements at different times are uncorrelated and each datum only depends on the state of the system in that moment. Then, the probabilistic observation operator $\mathcal{H}_i$ is defined as

$$\begin{gathered}\mathcal{H}_i : \mathcal{X}_i \to \mathcal{P}\left(\mathcal{Y}_i\right),\\ \mathcal{H}_i\left(x_i\right) := p_{\mathcal{H}_i\left(x_i\right)},\\ p_{\mathcal{H}_i\left(x_i\right)}\left(y_i\right) := p\left(y_i|x_i\right).\end{gathered} \tag{3.2}$$

Recalling the meaning of forecast and analysis in Section 2.2, the corresponding probabilities $p_i^f, p_i^a \in \mathcal{P}\left(\mathcal{X}_i\right)$ are

$$p_i^f\left(x_i\right) := p\left(x_i|y_{0:i-1}\right) \tag{3.3}$$

and

$$p_i^a\left(x_i\right) := p\left(x_i|y_{0:i}\right). \tag{3.4}$$

Both of them can be obtained in a sequential manner, known as Bayesian filter. As suggested by the name, a fundamental prerequisite is the Bayes Theorem, rewritten here for reader convenience in a very condensed form, along with a handily corollary:

Theorem 1 (Bayes Theorem)**.** *If A, B are two events in a probability space, then*

$$p\left(A|B\right) = \frac{p\left(B|A\right)p\left(A\right)}{p\left(B\right)}.$$

Corollary 2. *If A, B, C are three events in a probability space, then*

$$p\left(A|B,C\right) = \frac{p\left(B|A,C\right)p\left(A|C\right)}{p\left(B|C\right)}.$$

3.2 Bayesian Filter

Starting from definition (3.4), $p_i^a(x_i)$ can be rewritten as

$$p_i^a(x_i) = p\left(x_i|y_i, y_{0:i-1}\right) = \frac{p\left(y_i|x_i, y_{0:i-1}\right) p\left(x_i|y_{0:i-1}\right)}{p\left(y_i|y_{0:i-1}\right)}, \tag{3.5}$$

where the last equality comes from Bayes Theorem (Corollary 2).
Note that the denominator of equation (3.5) does not depend on x_i, then it is a constant number, that can be seen as a normalization factor of the distribution at the numerator. Using the symbol $\propto$ to indicate proportionality relation, equation (3.5) can be written as follows

$$p_i^a(x_i) \propto p\left(y_i|x_i, y_{0:i-1}\right) p\left(x_i|y_{0:i-1}\right). \tag{3.6}$$

Recalling that the datum y_i only depends on the state x_i and by definition (3.2), it holds that

$$p\left(y_i|x_i, y_{0:i-1}\right) = p\left(y_i|x_i\right) = p_{\mathcal{H}_i(x_i)}\left(y_i\right),$$

furthermore, by definition (3.3), equation (3.6) becomes

$$p_i^a\left(x_i\right) \propto p_{\mathcal{H}_i(x_i)}\left(y_i\right) p_i^f\left(x_i\right). \tag{3.7}$$

Now, p_i^f is obtained by the law of total probability:

Theorem 3 (Law of total probability)**.** *Let $\{A_j\}$ be a family of pairwise disjoint events. Then, for every event B, it holds that*

$$p\left(B\right) = \sum_j p\left(B|A_j\right) p\left(A_j\right).$$

Using Theorem 3 in definition (3.3),

$$p_i^f\left(x_i\right) = \int_{\mathbb{R}^N} p\left(x_i|x_{i-1}, y_{0:i-1}\right) p\left(x_{i-1}|y_{0:i-1}\right) dx_{i-1}$$

and finally, recalling that x_i and his uncertainty is fully determined by x_{i-1} and by definition (3.1)

$$p\left(x_i|x_{i-1}, y_{0:i-1}\right) = p\left(x_i|x_{i-1}\right) = p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right)$$

and by definition (3.4),

$$p_i^f\left(x_i\right) = \int_{\mathbb{R}^N} p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right) p_{i-1}^a\left(x_{i-1}\right) dx_{i-1}. \tag{3.8}$$

Summing up equations (3.8) and (3.7), knowing the analysis probability at time t_{i-1}, it is possible to calculate the forecast and analysis probability at time t_i via $\mathcal{M}_i$ and $\mathcal{H}_i$ by the sequential relation

$$\begin{cases} p_i^f\left(x_i\right) = \displaystyle\int_{\mathbb{R}^N} p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right) p_{i-1}^a\left(x_{i-1}\right) dx_{i-1}, \\ p_i^a\left(x_i\right) \propto p_{\mathcal{H}_i(x_i)}\left(y_i\right) p_i^f\left(x_i\right). \end{cases} \tag{3.9}$$

In spite of his formal elegance, equation (3.9) is impossible to be treated numerically as it is (at least for big N). Then, adding other hypothesis becomes necessary to reach a more viable expression. In particular, as presented in the next sections, the Kalman-Filter is obtained by assuming gaussian behaviours and linear operators.

3.3 Why gaussians

In order to derive the Kalman-Filter equations (as presented in data assimilation textbooks, e.g. [26]), it is useful to set some definitions and recall a few proprieties of Gaussian distributions.
Gaussians can be managed very easily, because they are completely defined by just mean and covariance and they interact "very well" with each other, as shown by Theorem 5, here presented after a preparatory definition and lemma.

If $x, a \in \mathbb{R}^d$ and A is a $d \times d$ real symmetric positive-definite matrix, let's indicate with $\mathcal{N}\left(x; a, A\right)$ the normal distribution of variable x, with mean a and covariance A, namely

$$\mathcal{N}\left(x; a, A\right) := \frac{1}{\sqrt{\left(2\pi\right)^d \left|A\right|}} \exp\left[-\frac{1}{2}\left(x-a\right)^T A^{-1}\left(x-a\right)\right],$$

where $|A|$ denotes the determinant of A.

Lemma 4. *Let A, B and M be real matrices of dimensions $n \times n$, $m \times m$ and $m \times n$ respectively. If A, B are symmetric and positive-definite, then the following equation holds*

$$\begin{aligned} &\left(Mx-y\right)^T A^{-1}\left(Mx-y\right) + x^T B^{-1} x = \\ &= \left(x - CM^TA^{-1}y\right)^T C^{-1}\left(x - CM^TA^{-1}y\right) + y^T\left(A + MBM^T\right)^{-1} y, \end{aligned} \tag{3.10}$$

where $x \in \mathbb{R}^n, y \in \mathbb{R}^m$ and

$$C := \left(M^TA^{-1}M + B^{-1}\right)^{-1}. \tag{3.11}$$

Proof. First of all note that A, B and C^{-1} are symmetric positive-definite, then they are invertible.
Now, due to the symmetry of A,

$$(Mx-y)^T A^{-1}(Mx-y) = x^T M^T A^{-1} Mx - 2x^T M^T A^{-1} y + y^T A^{-1} y,$$

and the left hand side of equation (3.10) becomes

$$\begin{aligned}(Mx-y)^T A^{-1}(Mx-y) + x^T B^{-1} x &= \\ &= x^T\left(M^T A^{-1} M + B^{-1}\right)x - 2x^T M^T A^{-1} y + y^T A^{-1} y. \end{aligned} \tag{3.12}$$

Using definition (3.11) and completing the square, it holds that

$$\begin{aligned} &x^T\left(M^T A^{-1} M + B^{-1}\right)x - 2x^T M^T A^{-1} y = \\ &= x^T C^{-1} x - 2x^T C^{-1} C M^T A^{-1} y \\ &= \left(x - CM^T A^{-1} y\right)^T C^{-1}\left(x - CM^T A^{-1} y\right) - y^T A^{-1} M C M^T A^{-1} y. \end{aligned} \tag{3.13}$$

Substituting equation (3.13) in equation (3.12) we obtain

$$\begin{aligned} &(Mx-y)^T A^{-1}(Mx-y) + x^T B^{-1} x = \\ &= \left(x - CM^T A^{-1} y\right)^T C^{-1}\left(x - CM^T A^{-1} y\right) + y^T A^{-1} y - y^T A^{-1} M C M^T A^{-1} y, \end{aligned} \tag{3.14}$$

where the last two terms can be rewritten as

$$y^T A^{-1} y - y^T A^{-1} M C M^T A^{-1} y = y^T\left(A^{-1} - A^{-1} M C M^T A^{-1}\right)y. \tag{3.15}$$

Since

$$\begin{aligned} &\left(A^{-1} - A^{-1} M C M^T A^{-1}\right)\left(A + MBM^T\right) = \\ &\quad = I_m + A^{-1} MBM^T - A^{-1} MCM^T - A^{-1} MCM^T A^{-1} MBM^T \\ &\quad = I_m + A^{-1} MBM^T - A^{-1} MCB^{-1} BM^T - A^{-1} MCM^T A^{-1} MBM^T \\ &\quad = I_m + A^{-1} MBM^T - A^{-1} MC\left(M^T A^{-1} M + B^{-1}\right) BM^T \\ &\quad = I_m, \end{aligned}$$

where I_m is the identity matrix of dimension m, then equation (3.15) can be written

$$y^T A^{-1} y - y^T A^{-1} M C M^T A^{-1} y = y^T\left(A + MBM^T\right)^{-1} y. \tag{3.16}$$

and, using equations (3.16) in (3.14), the lemma is proved. □

Theorem 5. *Let A, B and M be real matrices of dimensions $n \times n$, $m \times m$ and $m \times n$ respectively. If A, B are symmetric positive-definite matrices and $x \in \mathbb{R}^n, y \in \mathbb{R}^m$, then*

$$\mathcal{N}\left(y; Mx, A\right) \mathcal{N}\left(x; 0, B\right) = \mathcal{N}\left(y; 0, A + MBM^T\right) \mathcal{N}\left(x; CM^T A^{-1} y, C\right),$$

with

$$C = \left(M^T A^{-1} M + B^{-1}\right)^{-1}.$$

Proof. Omitting the normalization constants and using the proportional sign $\propto$, it holds that

$$\mathcal{N}\left(y; Mx, A\right) \mathcal{N}\left(x; 0, B\right) \propto \exp\left[-\frac{1}{2}\left(y - Mx\right)^T A^{-1}\left(y - Mx\right) - \frac{1}{2} x^T B^{-1} x\right].$$

By Lemma 4, it can be written

$$\begin{aligned} &\mathcal{N}\left(y; Mx, A\right) \mathcal{N}\left(x; 0, B\right) \propto \\ &\propto \exp\left[-\frac{1}{2} y^T (A + MBM^T)^{-1} y - \frac{1}{2}\left(x - CM^T A^{-1} y\right)^T C^{-1}\left(x - CM^T A^{-1} y\right)\right], \end{aligned}$$

and then

$$\mathcal{N}\left(y; Mx, A\right) \mathcal{N}\left(x; 0, B\right) \propto \mathcal{N}\left(y; 0, A + MBM^T\right) \mathcal{N}\left(x; CM^T A^{-1} y, C\right). \tag{3.17}$$

To check that equality holds, it sufficient to integrate both sides of expression (3.17) noting that they are already normalized:

$$\begin{aligned} &\int_{\mathbb{R}^n} \int_{\mathbb{R}^m} \mathcal{N}\left(y; Mx, A\right) \mathcal{N}\left(x; 0, B\right) dy dx = \\ &\qquad = \int_{\mathbb{R}^n} \mathcal{N}\left(x; 0, B\right) \int_{\mathbb{R}^m} \mathcal{N}\left(y; Mx, A\right) dy dx \\ &\qquad = \int_{\mathbb{R}^n} \mathcal{N}\left(x; 0, B\right) dx \\ &\qquad = 1 \end{aligned}$$

and

$$\begin{aligned}\int_{\mathbb{R}^n}\int_{\mathbb{R}^m} \mathcal{N}\left(y;0,A+MBM^T\right)\mathcal{N}\left(x;CM^TA^{-1}y,C\right)dydx = \\ &= \int_{\mathbb{R}^m}\int_{\mathbb{R}^n} \mathcal{N}\left(y;0,A+MBM^T\right)\mathcal{N}\left(x;CM^TA^{-1}y,C\right)dxdy \\ &= \int_{\mathbb{R}^m} \mathcal{N}\left(y;0,A+MBM^T\right)\int_{\mathbb{R}^n}\mathcal{N}\left(x;CM^TA^{-1}y,C\right)dxdy \\ &= \int_{\mathbb{R}^m} \mathcal{N}\left(y;0,A+MBM^T\right)dy \\ &= 1\end{aligned}$$

□

With Theorem 5, that is a powerful tool in order to manage Gaussian's interactions, we are ready to derive 3D-VAR and Kalman-Filter equations, as shown in the next two sections.

3.4 Bayesian derivation of the 3D-VAR

Starting from equation (3.9), rewritten here in an equivalent version but using a more general notation in order to show the background probability p^b_{i-1},

$$\begin{cases} p^b_{i-1}\left(x_{i-1}\right) = p^a_{i-1}\left(x_{i-1}\right) \\ p^f_i\left(x_i\right) = \int_{\mathbb{R}^N} p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right)p^b_{i-1}\left(x_{i-1}\right)dx_{i-1}, \\ p^a_i\left(x_i\right) \propto p_{\mathcal{H}_i(x_i)}\left(y_i\right)p^f_i\left(x_i\right), \end{cases} \tag{3.18}$$

we can simplify calculation by adding two assumptions and a (quite strong) approximation:

- Gaussian model error, or

$$p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right) = \mathcal{N}\left(x_i;m_i\left(x_{i-1}\right),Q_i\right), \tag{3.19}$$

- Gaussian observation error, or

$$p_{\mathcal{H}_i(x_i)}\left(y_i\right) = \mathcal{N}\left(y_i;h_i\left(x_i\right),R_i\right), \tag{3.20}$$

- deterministic background state, that is to say

$$p^b_{i-1}\left(x_{i-1}\right) = \delta_{x^b_{i-1}}\left(x_{i-1}\right), \tag{3.21}$$

where $\delta_{x^b_{i-1}}$ is the Dirac delta distribution[1] centred in x^b_{i-1}.

The last point means that there are no doubts on the state of the system at the previous time t_{i-1}. Obviously, this is a strong approximation in many fields and in particular in biogeochemical marine systems.
Substituting equations (3.19) and (3.21) in the second line of (3.18), it results

$$p^f_i\left(x_i\right) = \int_{\mathbb{R}^N} \mathcal{N}\left(x_i; m_i\left(x_{i-1}\right), Q_i\right) \delta_{x^b_{i-1}}\left(x_{i-1}\right) dx_{i-1} = \mathcal{N}\left(x_i; m_i\left(x^b_{i-1}\right), Q_i\right). \tag{3.22}$$

Using equations (3.20) and (3.22) in the third line of (3.18), it comes out that

$$p^a_i\left(x_i\right) \propto \mathcal{N}\left(y_i; h_i\left(x_i\right), R_i\right) \mathcal{N}\left(x_i; m_i\left(x^b_{i-1}\right), Q_i\right). \tag{3.23}$$

In order to obtain 3D-VAR equation, it is necessary to choose the mode of p^a_i as the estimation of x^a_i.
In fact we obtain, from equation (3.23),

$$\begin{aligned} x^a_i &= \operatorname{mode}\left(p^a_i\right) \\ &= \underset{x_i \in \mathbb{R}^N}{\operatorname{argmax}}\, p^a_i\left(x_i\right) \\ &= \underset{x_i \in \mathbb{R}^N}{\operatorname{argmax}}\, \mathcal{N}\left(y_i; h_i\left(x_i\right), R_i\right) \mathcal{N}\left(x_i; m_i\left(x^b_{i-1}\right), Q_i\right) \end{aligned}$$

and finally

$$x^a_i = \underset{x_i \in \mathbb{R}^N}{\operatorname{argmin}}\, J_i\left(x_i\right), \tag{3.24}$$

where

$$\begin{aligned} J_i\left(x_i\right) = \left(y_i - h_i\left(x_i\right)\right)^T R_i^{-1}\left(y_i - h_i\left(x_i\right)\right) + \\ + \left(x_i - m_i\left(x^b_{i-1}\right)\right)^T Q_i^{-1}\left(x_i - m_i\left(x^b_{i-1}\right)\right). \end{aligned} \tag{3.25}$$

Then, by using this analysis state as background in the next time step (approximating again the background probability to a Dirac delta and neglecting the uncertainties), i.e.

$$x^b_i = x^a_i,$$

the procedure is repeatable.
Since J_i in equation (3.25) is the 3D-VAR error functional appearing in equation (2.3), while equations (3.24) and (2.4) are the same too, then we have

[1] With a “little” notation abuse, for non-mathematician readers convenience, the Dirac delta is here treated like a probability density function, while it is a distribution instead (namely an element of the dual of the space of the test functions).

just obtained the 3D-VAR scheme with the Bayesian formalism.
This means that the hypothesis made in this section characterize the 3D-VAR, that, summarizing

- assumes normal errors on model and observations (with covariance matrices Q and R),
- uses the mode of the analysis probability distribution as estimation of the analysis state x_i^a
- and then, in the next step, uses x_i^a as new background state, neglecting any other information carried by p_i^a.

In particular, the second point presents a quite debatable choice, since the mean is usually a preferred estimator.
The last point instead is the main weakness of the 3D-VAR, representing a double drawback: the covariance information is not transferred to next time step and the uncertainty on the background state is neglected, leading to an overestimation of the confidence of the model prediction. For that reason, in 3D-VAR real implementations (e.g. [11], [44]), the Q_i model error covariance matrix is usually substituted by a much larger matrix B representing the system variability.

3.5 Derivation of the Kalman-Filter

In this section, the Kalman-Filter equations are obtained.
Compared with 3D-VAR derivation (Section 3.4), the Kalman-Filter overcomes all the weaknesses of the variational method by adding the hypothesis of normal background error. On the other hand, the price is the necessity of linear operators in order to keep the Gaussian behaviour.
Restarting from equation (3.18)

$$\begin{cases} p_{i-1}^b(x_{i-1}) = p_{i-1}^a(x_{i-1}) \\ p_i^f(x_i) = \displaystyle\int_{\mathbb{R}^N} p_{\mathcal{M}_i(x_{i-1})}(x_i)\, p_{i-1}^b(x_{i-1})\, dx_{i-1}, \\ p_i^a(x_i) \propto p_{\mathcal{H}_i(x_i)}(y_i)\, p_i^f(x_i), \end{cases} \tag{3.26}$$

Kalman-Filter equations can be obtained from five assumptions:

- Gaussian model error, or

$$p_{\mathcal{M}_i(x_{i-1})}(x_i) = \mathcal{N}(x_i; m_i(x_{i-1}), Q_i), \tag{3.27}$$

- Gaussian observation error, or
$$p_{\mathcal{H}_i(x_i)}\left(y_i\right)=\mathcal{N}\left(y_i;h_i\left(x_i\right),R_i\right), \tag{3.28}$$
- Gaussian background error, or
$$p_{i-1}^b\left(x_{i-1}\right)=\mathcal{N}\left(x_{i-1};x_{i-1}^b,P_{i-1}^b\right), \tag{3.29}$$
- linear model operator m_i, then it holds that
$$m_i\left(x_{i-1}\right)=m_i\left(x_{i-1}^b\right)+M_i\left(x_{i-1}-x_{i-1}^b\right), \tag{3.30}$$
where M_i is a $N\times N$ real matrix,
- linear observation operator h_i, then it holds that
$$h_i\left(x_i\right)=h_i\left(m_i\left(x_{i-1}^b\right)\right)+H_i\left(x_i-m_i\left(x_{i-1}^b\right)\right), \tag{3.31}$$
where H_i is a $n\times N$ real matrix.

Now, using equation (3.30) in (3.27), we obtain

$$p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right)=\mathcal{N}\left(x_i;m_i\left(x_{i-1}^b\right)+M_i\left(x_{i-1}-x_{i-1}^b\right),Q_i\right)$$

which becomes, after a simple change of variable,

$$p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right)=\mathcal{N}\left(x_i-m_i\left(x_{i-1}^b\right);M_i\left(x_{i-1}-x_{i-1}^b\right),Q_i\right). \tag{3.32}$$

Analogously, equation (3.29) can be written

$$p_{i-1}^b\left(x_{i-1}\right)=\mathcal{N}\left(x_{i-1}-x_{i-1}^b;0,P_{i-1}^b\right). \tag{3.33}$$

By equations (3.32), (3.33) and Theorem 5, it holds

$$\begin{aligned}
&p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right)p_{i-1}^b\left(x_{i-1}\right)=\\
&\quad=\mathcal{N}\left(x_i-m_i\left(x_{i-1}^b\right);M_i\left(x_{i-1}-x_{i-1}^b\right),Q_i\right)\mathcal{N}\left(x_{i-1}-x_{i-1}^b;0,P_{i-1}^b\right)\\
&\quad=\mathcal{N}\left(x_i-m_i\left(x_{i-1}^b\right);0,P_i^f\right)\cdot\\
&\qquad\qquad\cdot\mathcal{N}\left(x_{i-1}-x_{i-1}^b;P_{i-1}^pM_i^TQ^{-1}\left(x_i-m_i\left(x_{i-1}^b\right)\right),P_{i-1}^p\right),
\end{aligned} \tag{3.34}$$

where

$$P_i^f:=Q_i+M_iP_{i-1}^bM_i^T$$

and

$$P_{i-1}^p:=\left(M_i^TQ_i^{-1}M_i+\left(P_{i-1}^b\right)^{-1}\right)^{-1}. \tag{3.35}$$

Changing variable again, equation (3.34) becomes

$$p_{\mathcal{M}_i(x_{i-1})}\left(x_i\right) p_{i-1}^b\left(x_{i-1}\right) = \mathcal{N}\left(x_i; m_i\left(x_{i-1}^b\right), P_i^f\right) \mathcal{N}\left(x_{i-1}; x_{i-1}^p, P_{i-1}^p\right), \tag{3.36}$$

where

$$x_{i-1}^p \coloneqq x_{i-1}^b + P_{i-1}^p M_i^T Q_i^{-1}\left(x_i - m_i\left(x_{i-1}^b\right)\right), \tag{3.37}$$

and putting it inside the second line of equation (3.26) we obtain

$$p_i^f\left(x_i\right) = \mathcal{N}\left(x_i; m_i\left(x_{i-1}^b\right), P_i^f\right) \int_{\mathbb{R}^N} \mathcal{N}\left(x_{i-1}; x_{i-1}^p, P_{i-1}^p\right) dx_{i-1},$$

where the last integral is equal to 1 and vanishes.
Then, finally, the obtained forecast probability is a normal distribution such that

$$p_i^f\left(x_i\right) = \mathcal{N}\left(x_i; x_i^f, P_i^f\right), \tag{3.38}$$

with

$$x_i^f \coloneqq m_i\left(x_{i-1}^b\right). \tag{3.39}$$

Now, to compute analysis probability, a similar procedure can be adopted, so, using equations (3.28), (3.31), (3.38) and (3.39) it holds that

$$p_{\mathcal{H}_i(x_i)}\left(y_i\right) p_i^f\left(x_i\right) = \mathcal{N}\left(y_i; h_i\left(x_i^f\right) + H_i\left(x_i - x_i^f\right), R_i\right) \mathcal{N}\left(x_i; x_i^f, P_i^f\right). \tag{3.40}$$

Changing variable and using Theorem 5, it becomes

$$\begin{aligned} & p_{\mathcal{H}_i(x_i)}\left(y_i\right) p_i^f\left(x_i\right) = \\ & \quad = \mathcal{N}\left(y_i - h_i\left(x_i^f\right); H_i\left(x_i - x_i^f\right), R_i\right) \mathcal{N}\left(x_i - x_i^f; 0, P_i^f\right) \\ & \quad = \mathcal{N}\left(y_i - h_i\left(x_i^f\right); 0, P_i^l\right) \mathcal{N}\left(x_i - x_i^f; P_i^a H_i^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right), P_i^a\right), \end{aligned} \tag{3.41}$$

where

$$P_i^l \coloneqq R_i + H_i P_i^f H_i^T \tag{3.42}$$

and

$$P_i^a \coloneqq \left(H_i^T R_i^{-1} H_i + \left(P_i^f\right)^{-1}\right)^{-1}.$$

Changing variable in equation (3.41) and substituting it in the third line of equation (3.26), we obtain

$$p_i^a\left(x_i\right) \propto \mathcal{N}\left(y_i; h_i\left(x_i^f\right), P_i^l\right) \mathcal{N}\left(x_i; x_i^f + P_i^a H_i^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right), P_i^a\right). \tag{3.43}$$

Since the first term in the right hand side of the last expression is not depending on x_i, then it is just a constant factor and it is not relevant for the proportional relation. So,

$$p_i^a\left(x_i\right) \propto \mathcal{N}\left(x_i; x_i^f + P_i^a H_i^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right), P_i^a\right).$$

Finally, since the right hand side is already normalized, then the equality holds, and the analysis probability is

$$p_i^a\left(x_i\right) = \mathcal{N}\left(x_i; x_i^a, P_i^a\right),$$

with

$$x_i^a := x_i^f + P_i^a H_i^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right).$$

All together, Kalman-Filter equations are

$$\begin{cases} p_i^f\left(x_i\right) = \mathcal{N}\left(x_i; x_i^f, P_i^f\right), \\ p_i^a\left(x_i\right) = \mathcal{N}\left(x_i; x_i^a, P_i^a\right), \end{cases}$$

with

$$\begin{cases} P_i^f = Q_i + M_i P_{i-1}^b M_i^T, \\ x_i^f = m_i\left(x_{i-1}^b\right), \end{cases} \tag{3.44}$$

and

$$\begin{cases} P_i^a = \left(H_i^T R_i^{-1} H_i + \left(P_i^f\right)^{-1}\right)^{-1}, \\ x_i^a = x_i^f + P_i^a H_i^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right). \end{cases} \tag{3.45}$$

Obviously, in the next time step, the computed analysis mean and covariance can be used as background, i.e.

$$\begin{cases} P_i^b = P_i^a, \\ x_i^b = x_i^a. \end{cases}$$

Summarizing, the Kalman-Filter:

- starts from the same Gaussianity hypothesis of the 3D-VAR, adding background normal behaviour,
- needs linear operators,
- uses the mean as estimator,
- does not need any approximation to start the following time step.

In particular, the last two points mean that the Kalman-Filter is able to fully track the probability distributions involved at every time step, taking into account all the previous data without any loss of information.

Due to these features, Kalman-Filter approach seems very appealing compared with 3D-VAR. However, to make it viable in big complex systems, it is necessary to deal with two problems: hardly manageable huge covariance matrices and linear operators.

As a side note, it is interesting to observe that x_{i-1}^p, P_{i-1}^p and P_i^l, appearing in equations (3.35), (3.37) and (3.42), have a particular meaning (even if they are not used in the final Kalman-Filter expressions). The top right "p" is for "previous", and is related with propagation of probabilities backward in time, while "l" is for "likelihood", and quantifies how well data fit in the system. These subjects will be developed in details in Sections 5.2, 5.3 and 7.1.

Coming back to Kalman-Filter and summarizing, this method focus on Gaussian probabilities and tracks step by step the mean and the covariance matrix of the state, taking into account all the previous data without loss of information.

Chapter 4

From Kalman to SEIK

In geoscience, when working with forecasts in big complex systems, computational time is an important factor to take into account. Furthermore, the models usually are far from linearity.
As seen in Section 3.5, Kalman-Filter assumes linearity, and works with covariance matrices of side N, the dimension of the state vector, that can be of the order of 10^6 or even more. Matrices like that are near to be computationally untreatable, even for the most recent HPC systems.
Thus, it is necessary to stretch the Kalman-Filter hypothesis in order to not exclude non-linear systems, and find a way to make it computationally feasible. This chapter presents some Kalman-Filter evolution developed in literature to face these weaknesses.

4.1 Extended Kalman-Filter

Probably the first idea that comes to mind about non-linearity is to try linearising. The Extended Kalman-Filter method ([23]) substitute the linear assumption of Equations (3.30) and (3.31) with a linear approximation. Section 3.5 is already conveniently written to fit with this interpretation, and there are no differences in the derivation or in the final equations, a part from the meaning of M_i and H_i, that does no more exactly represent the respective operators. In fact, if M_i and H_i are considered the Jacobian matrices of m_i and h_i in the background and forecast states respectively, equations (3.30) and (3.31) becomes the first order Taylor expansion of model and observation operators around x^b_{i-1} and $m_i\left(x^b_{i-1}\right)$.
The idea of linearising around the background state (and its evolution) is an understandable decision as, hopefully, x_{i-1} should be quite near to its estimation x^b_{i-1} and the same should hold for x_i and $m_i\left(x^b_{i-1}\right)$.

However, since we are working with normal distributions, x_{i-1} can potentially be at an arbitrary distance from x^b_{i-1}, with a bigger or smaller probability depending on the form of P^b_{i-1}. Then, it can be possible that this linearisation fails to well approximate the operator.
This subject is better discussed in Section 4.5, and further developed in Section 5.1.
The Extended Kalaman-Filter then propose a possible solution to the linearity Kalman-Filter issue. In order to overcome the big dimension problem instead, a further improvement is needed. The following two sections provide useful elements to develop the SEIK filter, namely the evolution of the Extended Kalman-Filter capable of handle the big covariance matrices involved.

4.2 Dimensionality Reduction

The main idea behind the reduction of the dimension of the problem is that, often, the whole space $\mathcal{X}_i \cong \mathbb{R}^N$ is much larger than the subset of the system states with a realistic meaning. For example, in oceanography, the state where the whole ocean is at a temperature of $50°C$ it is not realistic (at least in a not apocalyptic scenario) and will never be computed, but it is still included in $\mathcal{X}_i$, "wasting dimensions".
The number of the realistic states is often much smaller compared with the whole space, and, mathematically speaking, they can be mapped, up to a certain precision, with a manifold of dimension $r \ll N$.
Without going too deeply into geometrical details, if a state x_{i-1} and its estimation x^b_{i-1} (both of them chosen inside the above r-dimensional manifold) are not too far one from each other, the error vector $x_{i-1} - x^b_{i-1}$ can be projected bijectively in a r-dimensional euclidean space $\mathbb{R}^r$.
Following this reasoning, the covariance matrix P^b_{i-1}, that "encodes" the information about which directions are more affected by the error, can be approximated by a rank r matrix by keeping only the information about the r more significant directions, or, equivalently, by projecting the error in a r-dimensional subspace.
At this purpose, next section will present (in a very fast and not detailed way) the Principal Component Analysis technique, that is the standard way to "decode" covariance matrices.

4.3 Principal Component Analysis

Principal Component Analysis (PCA) is a well known classical data analysis method (see for example [2]). This section includes a brief presentation of the relevant parts.
Given two independent scalar variables z_1 and z_2 with Gaussian behaviour,

$$p(z_1) = \mathcal{N}\left(z_1; a_1, {\sigma_1}^2\right),$$

$$p(z_2) = \mathcal{N}\left(z_2; a_2, {\sigma_2}^2\right),$$

their joint probability is obtained by the product of their density functions

$$p(z_1, z_2) = \mathcal{N}\left(z_1; a_1, {\sigma_1}^2\right) \mathcal{N}\left(z_2; a_2, {\sigma_2}^2\right).$$

The previous equation can easily be rewritten in vectorial form:

$$p(z) = \mathcal{N}(z; a, D),$$

where

$$z = \begin{pmatrix} z_1 \\ z_2 \end{pmatrix},$$

$$a = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix},$$

$$D = \begin{pmatrix} {\sigma_1}^2 & 0 \\ 0 & {\sigma_2}^2 \end{pmatrix}.$$

Thus, independent variables have a diagonal covariance matrix, and vice versa.
Starting from a general $N \times N$ covariance matrix P, the Principal Component Analysis method decomposes P in independent components, such that the first one has the maximum variance, the second one the maximum residual variance and so on so forth.
This can be obtained by the factorization

$$P = \Omega D \Omega^T$$

where Ω is an orthogonal matrix and D is a diagonal matrix with decreasing positive eigenvalues λ_i into the diagonal. Such decomposition always exists because P is symmetric and positive-definite.
In this way, using the change of variable

$$z = \Omega^T x,$$

the following equation holds:

$$\mathcal{N}\left(x; a, P\right) = \mathcal{N}\left(x; a, \Omega D \Omega^T\right) = \mathcal{N}\left(z; \Omega^T a, D\right),$$

and then

$$\mathcal{N}\left(x; a, P\right) = \mathcal{N}\left(z_1; b_1, {\sigma_1}^2\right) \cdot \ldots \cdot \mathcal{N}\left(z_N; b_N, {\sigma_N}^2\right), \tag{4.1}$$

with

$${\sigma_i}^2 = \lambda_i, \forall i \in \{1, \ldots, N\}$$

and

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_N \end{pmatrix} = \Omega^T a.$$

Thus, the columns of Ω represents an orthonormal system of coordinates that can be used to split the N-dimensional Gaussian distribution associated to P in N independent scalar normal variables, with decreasing variance.
Now, in order to reduce the dimension of the Gaussian at the left hand side of equation (4.1), we can forget the uncertainty of the last $N - r$ variables at the right hand side (the ones with the smallest variance), substituting their normal distributions with the limit for σ_i going to 0, namely the Dirac delta $\delta_{b_i}\left(z_i\right)$.
That is to say that

$$\mathcal{N}\left(x; a, P\right) \approx \mathcal{N}\left(\tilde{z}; \tilde{b}, A\right),$$

where the N-dimensional Gaussian at the left had side is approximated with an r-dimensional one via the embedding

$$\begin{array}{ccc} \mathbb{R}^r & \longrightarrow & \mathbb{R}^N \\ \tilde{z} & \longmapsto & x = L\tilde{z}, \end{array}$$

with

$$\tilde{z} = \begin{pmatrix} z_1 \\ \vdots \\ z_r \end{pmatrix},$$

$$\tilde{b} = \begin{pmatrix} b_1 \\ \vdots \\ b_r \end{pmatrix},$$

$$A = \begin{pmatrix} {\sigma_1}^2 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & {\sigma_r}^2 \end{pmatrix}$$

and L is the $N \times r$ matrix made by the first r columns of Ω.
Analogously, it can be written

$$\mathcal{N}\left(x; a, P\right) \approx \mathcal{N}\left(x; a, LAL^T\right),$$

where, forgiving the more permissive notation, the right hand side covariance matrix is a $N \times N$ singular matrix of rank r.
Thanks to this approximation, most of the costly operation can be projected and easily executed in the r-dimensional space, as long as r is small enough. In particular, next section presents how to manage the huge Extended Kalman-Filter matrices, starting by a decomposition that can be eventually obtained by the PCA method.

4.4 The SEEK Filter

The Singular Evolutive Extended Kalman-Filter ([38]), or SEEK, responds to the needing of a computationally feasible version of the Extended Kalman-Filter. Thus, instead of working with a full rank background covariance matrix, it uses a singular rank r matrix in the decomposed form

$$P^b_{i-1} = L_{i-1} A^b_{i-1} L^T_{i-1}, \tag{4.2}$$

where A^b_{i-1} and L_{i-1} are a $r \times r$ and a $N \times r$ full rank matrices, with A^b_{i-1} symmetric positive-definite.
By equation (4.2), the Extended Kalman-filter forecast covariance matrix in the first line of equations (3.44) becomes

$$P^f_i = Q_i + M_i L_{i-1} A^b_{i-1} L^T_{i-1} M^T_i = Q_i + L_i A^b_{i-1} L^T_i, \tag{4.3}$$

where

$$L_i = M_i L_{i-1}.$$

If M_i is not singular (which is a very weak request in geoscience models, because processes can usually be integrated backward in time), then L_i keeps the rank of L_{i-1} and its columns are a system of linearly independent vectors that generates an r-dimensional subspace called $\mathcal{L} \subseteq \mathbb{R}^N$.

Now, to keep working with rank r matrices, it is necessary to approximate Q.
At this purpose, Pham propose ([38]) to use

$$Q_i \approx L_i \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} L_i^T,$$

that represents an orthogonal projection sending the model error into the subspace $\mathcal{L}$, as can be proved by Corollary 9, here presented after some preparatory results. A reader not interested into mathematical details can go directly to equation (4.14).

Lemma 6. *Let U, V and A be real matrices of dimensions $n \times m$, $n \times l$ and $n \times n$ respectively. If A is symmetric and positive-definite and*

$$\Omega = \left(\begin{array}{c|c} U & V \end{array} \right) \tag{4.4}$$

is an orthogonal matrix, then the following equation holds

$$\begin{aligned} &(Ux + Vy)^T A^{-1} (Ux + Vy) = \\ &\qquad = \left(x + BU^T A^{-1} V y\right)^T B^{-1} \left(x + BU^T A^{-1} V y\right) + y^T \left(V^T A V\right)^{-1} y, \end{aligned}$$

where $x \in \mathbb{R}^m, y \in \mathbb{R}^l$ and

$$B := \left(U^T A^{-1} U\right)^{-1}. \tag{4.5}$$

Proof. Due to the symmetry of A,

$$(Ux + Vy)^T A^{-1} (Ux + Vy) = x^T U^T A^{-1} U x + 2x^T U^T A^{-1} V y + y^T V^T A^{-1} V y, \tag{4.6}$$

Using definition (4.5) and completing the square, it holds that

$$\begin{aligned} &x^T U^T A^{-1} U x + 2x^T U^T A^{-1} V y = x^T B^{-1} x + 2x^T B^{-1} B U^T A^{-1} V y \\ &= \left(x + BU^T A^{-1} V y\right)^T B^{-1} \left(x + BU^T A^{-1} V y\right) - y^T V^T A^{-1} U B U^T A^{-1} V y. \end{aligned} \tag{4.7}$$

Using equation (4.6) in (4.7), to complete the proof it is sufficient to prove that

$$V^T A^{-1} V - V^T A^{-1} U B U^T A^{-1} V = \left(V^T A V\right)^{-1}$$

or, equivalently,

$$V^T A^{-1} V V^T A V - V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} V V^T A V = I_n \tag{4.8}$$

From hypothesis (4.4),

$$U^T V = 0 \tag{4.9}$$

and

$$I_n = \Omega\Omega^T = UU^T + VV^T \tag{4.10}$$

or

$$VV^T = I_n - UU^T. \tag{4.11}$$

Using equation (4.11), the second term in the left hand side of equation (4.8) becomes

$$\begin{aligned}
V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} V V^T A V = & \\
&= V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} A V - \\
&\quad - V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} U U^T A V \\
&= V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T V - \\
&\quad - V^T A^{-1} U U^T A V
\end{aligned}$$

and finally, by equation (4.9),

$$V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} V V^T A V = -V^T A^{-1} U U^T A V. \tag{4.12}$$

Then, equation (4.8) can be proved using equations (4.12) and (4.10),

$$\begin{aligned}
V^T A^{-1} V V^T A V - V^T A^{-1} U \left(U^T A^{-1} U\right)^{-1} U^T A^{-1} V V^T A V = & \\
&= V^T A^{-1} V V^T A V + V^T A^{-1} U U^T A V = V^T A^{-1} \left(V V^T + U U^T\right) A V \\
&= V^T V = I_n,
\end{aligned}$$

where last equivalence comes from hypothesis (4.4). □

Corollary 7. *Let U and V be matrices of dimensions $n \times m$ and $n \times l$ respectively. Furthermore, let A, P be symmetric positive-definite $n \times n$ matrices such that*

$$\Omega^T P \Omega = I_n,$$

where

$$\Omega = \left(\begin{array}{c|c} U & V \end{array} \right).$$

Then the following equation holds

$$\begin{aligned}
(Ux + Vy)^T A^{-1} (Ux + Vy) = & \\
&= \left(x + BU^T A^{-1} V y\right)^T B^{-1} \left(x + BU^T A^{-1} V y\right) + y^T \left(V^T P A P V\right)^{-1} y,
\end{aligned}$$

where $x \in \mathbb{R}^m, y \in \mathbb{R}^l$ and

$$B := \left(U^T A^{-1} U\right)^{-1}.$$

Proof. Since P is symmetric positive-definite, Silvester's Theorem ensures that exists a decomposition

$$P = C^T C,$$

for some $n \times n$ matrix C.
Thus,

$$U' := CU,$$
$$V' := CV,$$

and

$$A' := CAC^T$$

satisfy the hypothesis of Lemma 6, that can be used to obtain the thesis. In fact

$$\begin{aligned}
&(Ux + Vy)^T A^{-1} (Ux + Vy) = \\
&\quad = (Ux + Vy)^T C^T \left(C^T\right)^{-1} A^{-1} C^{-1} C (Ux + Vy) \\
&\quad = (U'x + V'y)^T A'^{-1} (U'x + V'y) \\
&\quad = \left(x + BU'^T A'^{-1} V'y\right)^T B^{-1} \left(x + BU'^T A'^{-1} V'y\right) + y^T \left(V'^T A' V'\right)^{-1} y \\
&\quad = \left(x + BU^T A^{-1} Vy\right)^T B^{-1} \left(x + BU^T A^{-1} Vy\right) + y^T \left(V^T PAPV\right)^{-1} y,
\end{aligned}$$

with

$$B = \left(U'^T A'^{-1} U'\right)^{-1} = \left(U^T A^{-1} U\right)^{-1}.$$

□

Theorem 8. *Let U and V be matrices of dimensions $n \times m$ and $n \times l$ respectively. Furthermore, let A, P be symmetric positive-definite $n \times n$ matrices such that*

$$\Omega^T P \Omega = I_n,$$

where

$$\Omega = \left(\begin{array}{c|c} U & V \end{array} \right).$$

Then

$$\begin{aligned}
&\mathcal{N}(Ux + Vy; Ua + Vb, A) = \\
&\qquad\qquad = \mathcal{N}\left(x; a - BU^T A^{-1} V (y - b), B\right) \mathcal{N}\left(y; b, V^T PAPV\right),
\end{aligned}$$

where $x, a \in \mathbb{R}^m, y, b \in \mathbb{R}^l$ and

$$B := \left(U^T A^{-1} U\right)^{-1}.$$

Proof. It comes easily from Corollary 7. □

Corollary 9. *The projection of a Gaussian is a Gaussian. Furthermore, if $x, a \in \mathbb{R}^m$ and $y, b \in \mathbb{R}^l$ are real vectors, U and V are matrices of dimensions $n \times m$ and $n \times l$ respectively, and A, P are symmetric positive-definite $n \times n$ matrices such that*

$$\Omega^T P \Omega = I_n,$$

where

$$\Omega = \left(\begin{array}{c|c} U & V \end{array} \right),$$

then the projection of the Gaussian $\mathcal{N}(Ux + Vy; Ua + Vb, A)$ into the subspace generated by the columns of V, along the direction parallel to the subspace generated by the columns of U, is $\mathcal{N}\left(y; b, V^T PAPV\right)$.

Proof. Starting from Theorem 8, it is sufficient to integrate over x. □

Now we can use Corollary 9 to compute an approximation of Q through its orthogonal projection into $\mathcal{L}$. To do so, let V be an $N \times r$ matrix with orthonormal columns that are a base for $\mathcal{L}$, thus

$$L_i = VB \tag{4.13}$$

for some $r \times r$ invertible matrix B.
Such base can be completed to become an orthonormal base of $\mathbb{R}^N$, and let U be the matrix containing the missing $N - r$ vectors, so

$$\Omega = \left(\begin{array}{c|c} U & V \end{array} \right),$$

with Ω an $N \times N$ orthogonal matrix.
Then, the projection of the model error into $\mathcal{L}$ is an r-dimensional Gaussian with covariance $V^T Q_i V$ and, namely

$$\mathcal{N}\left(\tilde{z}; 0, V^T Q_i V\right),$$

where $z \in \mathbb{R}^r$ are the coordinates in the V base.
Coming back in N dimensions by the V matrix, Q_i can be approximated as

$$Q_i \approx VV^T Q_i VV^T = L_i B^{-1} \left(B^T\right)^{-1} L_i^T Q_i L_i B^{-1} \left(B^T\right)^{-1} L_i^T,$$

where last equality comes from equation (4.13).
Finally, by the orthonormality of the columns of V and again from equation (4.13), we have

$$Q_i \approx L_i \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} L_i^T. \tag{4.14}$$

Before to continue, note that this approximation is potentially very distant from the true value of Q, since there are no reasons for the model error to be mainly included into $\mathcal{L}$. Handling this problem is part of the original work of this thesis and it will be discussed in details in Section 5.2.
Let's now proceed by using (4.14) in equation (4.3), obtaining

$$P_i^f \approx L_i \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} L_i^T + L_i A_{i-1}^b L_i^T = L_i A_i^f L_i^T,$$

with

$$A_i^f := \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} + A_{i-1}^b. \tag{4.15}$$

Said with a probabilistic notation, it is

$$p_i^f \left(x_i\right) = \mathcal{N}\left(x_i; x_i^f, P_i^f\right) \approx \mathcal{N}\left(z; 0, A_i^f\right), \tag{4.16}$$

where $z \in \mathbb{R}^r$ are the coordinates in the base L_i, i.e.

$$\begin{array}{rcl} \mathbb{R}^r & \longrightarrow & \mathbb{R}^N \\ z & \longmapsto & x_i = L_i z + x_i^f. \end{array} \tag{4.17}$$

Now, using the approximation in equation (4.16) and the change of variable (4.17) in equation (3.40), it becomes

$$p_{\mathcal{H}_i(x_i)}\left(y_i\right) p_i^f \left(x_i\right) = \mathcal{N}\left(y_i; h_i\left(x_i^f\right) + H_i L_i z, R_i\right) \mathcal{N}\left(z; 0, A_i^f\right)$$

and, using Theorem 5 as done in Section 3.5, equation (3.43) becomes

$$p_i^a \left(x_i\right) \propto \mathcal{N}\left(y_i; h_i\left(x_i^f\right), A_i^l\right) \mathcal{N}\left(z; A_i^a \left(H_i L_i\right)^T R_i^{-1} \left(y_i - h_i\left(x_i^f\right)\right), A_i^a\right),$$

where

$$A_i^l := R_i + H_i L_i A_i^f \left(H_i L_i\right)^T$$

and

$$A_i^a := \left(\left(H_i L_i\right)^T R_i^{-1} H_i L_i + \left(A_i^f\right)^{-1}\right)^{-1}.$$

After a normalization and using embedding (4.17) to come back to $\mathbb{R}^N$, we have

$$p_i^a \left(x_i\right) \approx \mathcal{N}\left(x_i; x_i^f + L_i A_i^a \left(H_i L_i\right)^T R_i^{-1} \left(y_i - h_i\left(x_i^f\right)\right), L_i A_i^a L_i^T\right).$$

Then, summarizing, the SEEK reduces the dimension of the problem, approximating the covariance with a low rank matrices that can be easily managed in term of a base L_i and a reduced dimension covariance A_i.
The final equations are

$$\left\{ \begin{array}{l} p_i^f\left(x_i\right) \approx \mathcal{N}\left(x_i; x_i^f, L_i A_i^f L_i^T\right), \\ p_i^a\left(x_i\right) \approx \mathcal{N}\left(x_i; x_i^a, L_i A_i^a L_i^T\right), \end{array} \right.$$

with

$$\left\{ \begin{array}{l} L_i = M_i L_{i-1}, \\ A_i^f = \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} + A_{i-1}^b, \\ x_i^f = m_i\left(x_{i-1}^b\right), \end{array} \right. \tag{4.18}$$

and

$$\left\{ \begin{array}{l} A_i^a = \left(\left(H_i L_i\right)^T R_i^{-1} H_i L_i + \left(A_i^f\right)^{-1}\right)^{-1}, \\ x_i^a = x_i^f + L_i A_i^a \left(H_i L_i\right)^T R_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right). \end{array} \right. \tag{4.19}$$

Hugely reducing the computational cost of the Extended Kalman-Filter, the SEEK-Filter is able to overcome both the weaknesses of the Kalman-Filter, namely its linearity and big dimension matrices.
However new problems appeared. First of all, linearisation can lead to big errors in chaotic systems with high uncertainty, as pointed in Section 4.1.
Secondly, the approximation of the model error by equation (4.14) is far from sharpness.
Lastly, M_i and H_i, the Jacobian matrices of the operators, are needed at every step of the algorithm and their computation, in particular of M_i, can be quite burdening.
The next section propose a solution for the first and the last of the above SEEK weaknesses. The second one instead will be discussed in details later in Section 5.2.

4.5 Ensemble Data Assimilation and the SEIK Filter

Extended Kalman-Filter and SEEK, when used in fast changing chaotic systems, can lead to big errors and even divergences (see [30]).
A simple example can help to clarify the situation. If the model operator m_i changes slowly around the centre x_{i-1}^b of the linearisation, then the linear approximation fails only if x_{i-1} is far from x_{i-1}^b. If the variance σ^2 of the model error is small, then the probability of being far from the background state

decreases exponentially with distance, taking under control the linearisation error.
On the other hand, in case of higher uncertainty (i.e. big σ^2), the probability of being far enough is not negligible and, if the model has a more chaotic behaviour and the approximation works well only in a very small region, then the linearisation error can explode.
In order to avoid similar situations, the Singular Evolutive Interpolated Kalman-Filter ([37]), or SEIK, instead of approximating the operator m_i, approximate directly the Gaussian distribution with a discrete probability function. The ensemble of points with non-zero probability can be evolved via the m_i operator without the need of a linearisation.
Before entering into the details of SEIK, it is useful to spend a few words on the concept of Ensemble Data Assimilation. This topic encompass all those strategies that use, evolve and operate over a set of system states, instead of just one, in order to obtain a better estimation of the real state of the system.
The improvements in information technologies and the development of large clusters of computers for parallel computation acted as propellent for ensemble methods, that are naturally suited for parallelization. For this reason, a lot of interest has grown around ensembles in the last decades as well as the number of algorithms based on them. In particular, both SEEK and SEIK filters are ensemble methods, and well fit with the modern calculus infrastructures.
To understand why SEEK is considered an ensemble filter, it is sufficient to observe the first line of equation (4.18), i.e.

$$L_i = M_i L_{i-1}. \tag{4.20}$$

The matrices L_{i-1} and L_i contain the base vectors of the error subspace at time t_{i-1} and t_i respectively. Then equation (4.20) can be rewritten

$$l_i^j = M_i l_{i-1}^j, \forall j \in \{1, \dots, r\}, \tag{4.21}$$

with

$$L_{i-1} = \left(\begin{array}{c|c|c} l_{i-1}^1 & \cdots & l_{i-1}^r \end{array} \right)$$

and

$$L_i = \left(\begin{array}{c|c|c} l_i^1 & \cdots & l_i^r \end{array} \right).$$

Adding the third line of equation (4.18) to equation (4.21) we obtain

$$x_i^f + l_i^j = m_i\left(x_{i-1}^b\right) + M_i l_{i-1}^j, \forall j \in \{1, \ldots, r\},$$

that can be written, for every $j \in \{1, \ldots, r\}$,

$$x_i^j = m_i\left(x_{i-1}^b\right) + M_i\left(x_{i-1}^j - x_{i-1}^b\right), \tag{4.22}$$

where

$$x_{i-1}^j := x_{i-1}^b + l_{i-1}^j$$

and

$$x_i^j := x_i^f + l_i^j.$$

This means that $\left\{x_{i-1}^1, \ldots, x_{i-1}^r\right\}$ is an ensemble of states picked around the background state (that is to say, into the r-dimensional affine subspace built around x_{i-1}^b and containing all the possible values of x_{i-1}), that evolve via the linearisation of m_i (as can be seen in equation (4.22)) to obtain the ensemble $\{x_i^1 \ldots, x_i^r\}$ of states around the forecast state.
Thus, instead of using equation (4.20), the L_i matrix can be built by evolving a certain ensemble of states $\left\{x_{i-1}^1, \ldots, x_{i-1}^r\right\}$.
The SEIK algorithm works in a similar manner but, differently from SEEK, it uses directly m_i, instead of its linear approximation, over an ensemble of wisely chosen states.
Going deeper into SEIK details, it is important to note that the following SEIK presentation is an original reworked and expanded version of the classical algorithm, that it is included as a particular case (namely by choosing identical weights on the second order exact sampling procedure). The advantage is that, in this form, the algorithm is already prepared for the modifications presented in the next chapter.
That said, the starting point is the same as SEEK, namely

$$P_{i-1}^b = L_{i-1} A_{i-1}^b L_{i-1}^T. \tag{4.23}$$

A_{i-1}^b is the covariance matrix of the background error (in the reduced r-dimensional subspace) and, by Sylvester's Theorem, it can be decomposed as

$$A_{i-1}^b = CC^T, \tag{4.24}$$

for some real $r \times r$ matrix C.
Then, the equation

$$A_{i-1}^b = CU^T UC^T,$$

remains true for any $\tilde{r} \times r$ matrix U, as long as

$$U^T U = I_r. \tag{4.25}$$

Thus, U must have orthonormal columns and $\tilde{r} \geq r$. Finally, if

$$v := \begin{pmatrix} v_1 \\ \vdots \\ v_{\tilde{r}} \end{pmatrix} \in \mathbb{R}^{\tilde{r}}$$

and

$$w := \begin{pmatrix} {v_1}^2 \\ \vdots \\ {v_{\tilde{r}}}^2 \end{pmatrix} \in \mathbb{R}^{\tilde{r}}, \tag{4.26}$$

then

$$A^b_{i-1} = CU^T V^{-1} W V^{-1} U C^T, \tag{4.27}$$

where

$$V := \begin{pmatrix} v_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & v_{\tilde{r}} \end{pmatrix} \tag{4.28}$$

and

$$W := \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & w_{\tilde{r}} \end{pmatrix}.$$

Equation (4.27) can be used to sample an ensemble of weighted points that has zero mean and covariance A^b_{i-1} by choosing v and U such that

$$v^T v = 1 \tag{4.29}$$

and

$$U^T v = 0, \tag{4.30}$$

or, equivalently,

$$\left(\begin{array}{c|c} v & U \end{array} \right) = \Omega, \tag{4.31}$$

where Ω is a $\tilde{r} \times (r+1)$ matrix with orthonormal columns.
In fact, if the ensemble is made by the $\tilde{r}$ columns of the $r \times \tilde{r}$ matrix $\tilde{C}$, defined as

$$\tilde{C} := CU^T V^{-1}, \tag{4.32}$$

and each ensemble member has is probability weight defined by the coordinates of w (that sums to one due to equations (4.29) and (4.26)), then the weighted ensemble mean is

$$\tilde{C}w = CU^T V^{-1} w = CU^T v = 0, \tag{4.33}$$

where the last equivalences come from equations (4.32), (4.28), (4.26) and (4.30).
Thus, by equations (4.27) and (4.32), the ensemble covariance is

$$\tilde{C}W\tilde{C}^T = A^b_{i-1}. \tag{4.34}$$

In order to minimize complexity, the number of ensemble members is minimized, that is to say $\tilde{r} = r + 1$, as can be seen from equation (4.31), where Ω becomes a square orthogonal matrix.
This sampling technique is called minimum second order exact sampling ([37]) and this subject will be further developed in Section 5.1.
Now, the obtained ensemble can be embedded into the N-dimensional state space as usual by L_{i-1}, obtaining the Z_{i-1} matrix

$$Z_{i-1} = L_{i-1}\tilde{C}, \tag{4.35}$$

the columns of which have zero weighted mean (by equations (4.33) and (4.35)) and covariance P^b_{i-1} (by equations (4.34) and (4.23)).
Finally, by adding to each column vector z^j_{i-1} of Z_{i-1} the background state x^b_{i-1}, we have an ensemble of weighted states that, having same mean and covariance, approximates p^b_{i-1}.
Then, the evolution of the background probability can be approximated by the evolution of the ensemble: if X_i is the matrix having the evolved ensemble members as columns, i.e.

$$X_i = \left(\begin{array}{c|c|c} m_i\left(x^b_{i-1} + z^1_{i-1}\right) & \cdots & m_i\left(x^b_{i-1} + z^{r+1}_{i-1}\right) \end{array} \right),$$

the ensemble mean (which is chosen to represent the forecast state) is

$$x^f_i = X_i w, \tag{4.36}$$

while the covariance matrix is

$$Z_i W Z_i^T, \tag{4.37}$$

where, calling $\mathbf{1}$ the filled-by-ones matrix and by equation (4.36),

$$Z_i := X_i - x_i^f \mathbf{1}_{1\times(r+1)} = X_i \left(I_{r+1} - w\mathbf{1}_{1\times(r+1)} \right)$$

is the $N \times (r+1)$ matrix with columns spanning the r-dimensional error subspace in $\mathbb{R}^N$ (in the general case, while some very unlikely corner cases can produce a smaller dimension). Since $Z_i w = 0$, the matrix is not full rank, and the first r columns are sufficient to be a base of the above plane, while the last one can be expressed as function of the previous ones. If L_i is the new base matrix, then

$$L_i = Z_i \left(\begin{array}{c} I_r \\ \hline 0 \quad \cdots \quad 0 \end{array} \right) = X_i T, \tag{4.38}$$

with

$$T := \left(\begin{array}{c} I_r \\ \hline 0 \quad \cdots \quad 0 \end{array} \right) - w\mathbf{1}_{1\times r}. \tag{4.39}$$

Noting that $L_i = Z_i T$, if T^* is left inverse of T, with the same property of Z_i, namely $T^* w = 0$, then $Z_i = L_i T^*$. Since

$$\left(T^T W^{-1} T\right)^{-1} T^T W^{-1} T = I_r$$

and

$$T^T W^{-1} w = 0,$$

then

$$T^* = \left(T^T W^{-1} T\right)^{-1} T^T W^{-1}$$

and

$$Z_i = L_i \left(T^T W^{-1} T\right)^{-1} T^T W^{-1}.$$

Finally, the covariance of equation (4.37) can be written

$$Z_i W Z_i^T = L_i \left(T^T W^{-1} T\right)^{-1} T^T W^{-1} T \left(T^T W^{-1} T\right)^{-1} L_i^T = L_i \left(T^T W^{-1} T\right)^{-1} L_i^T.$$

Now it is possible to proceed as done in equations (4.3) and (4.15), obtaining

$$P_i^f = Q_i + L_i \left(T^T W^{-1} T\right)^{-1} L_i^T \tag{4.40}$$

and

$$A_i^f := \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} + \left(T^T W^{-1} T\right)^{-1}. \tag{4.41}$$

In order to obtain the analysis state and its covariance, SEEK algorithm used 4.19, that involves the Jacobian matrix H_i. In the SEIK scheme instead, the observation operator h_i is interpolated by using the ensemble members. Thus, the H_iL_i product in equation (4.19) is replaced by Y_iT, where Y_i is the matrix obtained by applying h_i to the columns of X_i.
Summarizing, SEIK equations read

$$\begin{cases} p_i^f\left(x_i\right) \approx \mathcal{N}\left(x_i; x_i^f, L_iA_i^fL_i^T\right), \\ p_i^a\left(x_i\right) \approx \mathcal{N}\left(x_i; x_i^a, L_iA_i^aL_i^T\right), \end{cases}$$

with

$$\begin{cases} L_i = X_iT, \\ A_i^f = \left(L_i^TL_i\right)^{-1}L_i^TQ_iL_i\left(L_i^TL_i\right)^{-1} + \left(T^TW^{-1}T\right)^{-1}, \\ x_i^f = X_iw, \end{cases}$$

and

$$\begin{cases} A_i^a = \left(\left(Y_iT\right)^TR_i^{-1}Y_iT + \left(A_i^f\right)^{-1}\right)^{-1}, \\ x_i^a = x_i^f + L_iA_i^a\left(Y_iT\right)^TR_i^{-1}\left(y_i - h_i\left(x_i^f\right)\right), \end{cases}$$

where X_i and Y_i are obtained using m_i and h_i on the ensemble extracted from p_{i-1}^b by minimum second order exact sampling.
The use of this sampling technique is the main and most important difference between SEIK and SEEK.
In fact, as proved by Theorem 12 in the next chapter, this produce a second order approximation of m_i, resulting in a better convergence compared with SEEK's linearisations.
Note that, differently from m_i, h_i is only interpolated by the previously used ensemble, without a dedicated exact sampling procedure.

Chapter 5

Beyond SEIK

The SEIK filter presented in the previous chapter has the good properties that we were looking for. In fact it joins the advantages of the Kalman-Filter's Bayesian approach, greatly avoiding its weaknesses, namely the linearity and the big dimension problem. This chapter is dedicated to present the original work made to further improve the skill of the SEIK filter.
First of all, a new sampling method with a higher order of convergence has been derived.
This has been achieved through the following steps:

- the second order exact sampling is investigated in deeper details and extended to higher orders,
- increasing the order of convergence results in a higher number of ensemble members (augmenting the computational cost), thus it is necessary to avoid this drawback,
- the new sampling strategies is obtained by building a minimum second order exact sample with the following property: the projection of the ensemble in the lower dimension subspace spanned by the most relevant PCA components has a higher order of convergence in that subspace.

Then, particular attention will be paid to the model error subject. Nowadays, various techniques, falling under the name of "inflation", are used to partially take into account model uncertainties. In contrast, some substantial modifications to SEEK and SEIK algorithms will be presented, obtaining a filter that accounts for the whole model error, splitting it at every step into a relevant and a noise-like components. The former will be taken into account in the forecast, while the latter will be treated as representativeness error and used to correct the measurement error.

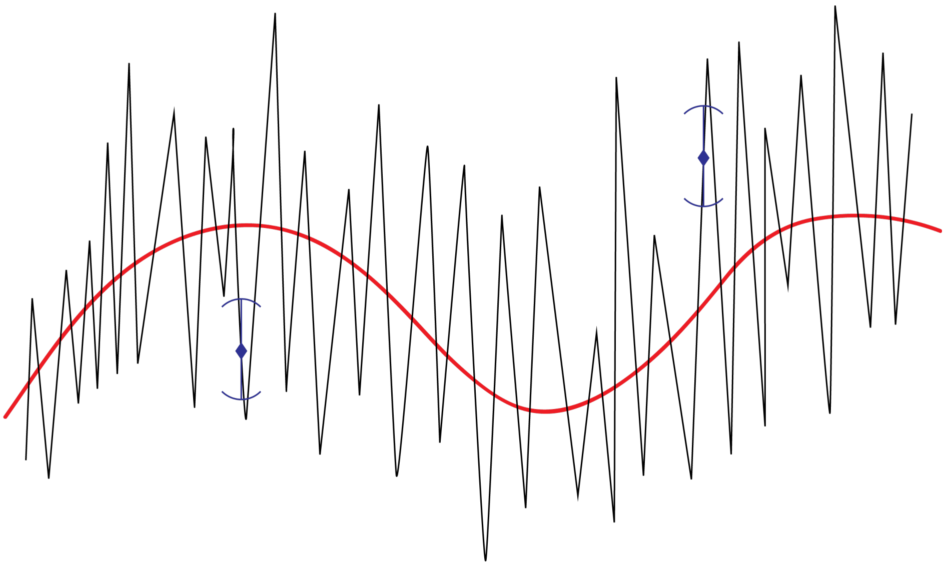

Figure 5.1: Model error and observations. Red line is the model trajectory. Black line is the true state. Blu dots are observation.

Figure 5.1 shows how the neglected components of the model error can lead to a representativeness error. The blue dots represents the observations: even if they are picked near to the reality (black line), they are still far from the average behaviour evolved by the model.
Finally, a maximum likelihood approach will be used to estimate the model error directly from data.

5.1 Higher order exact sampling

This section is dedicated to the second order exact sampling technique, and its generalization.It will be shown that a 2γ-th order exact sampling is able to approximate the mean of a distribution up to the 2γ-th order and its variance up to the γ-th order.
Finally, this will be used to build a minimum second order ensemble with growing order in the most relevant directions.
Let's start by defining $\mu_\gamma^{j_1,\dots,j_\gamma}$, the centred moment tensor of order γ of a random variable $x \in \mathbb{R}^n$ with probability density function $p(x)$:

$$\forall j_1,\dots,j_\gamma \in \{1,\dots,n\}, \mu_\gamma^{j_1,\dots,j_\gamma} := \int_{\mathbb{R}^n} (x_{j_1} - \bar{x}_{j_1}) \cdots \left(x_{j_\gamma} - \bar{x}_{j_\gamma}\right) p(x)\, dx, \tag{5.1}$$

with

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

and

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} = \operatorname{mean} x = \int_{\mathbb{R}^n} xp\left(x\right) dx.$$

Analogously, if x has a discrete distribution with weights $w_1, \ldots, w_r$ on the points $x^1, \ldots, x^r$, then

$$\forall j_1, \ldots, j_\gamma \in \left\{1, \ldots, n\right\}, \mu_\gamma^{j_1,\ldots,j_\gamma} = \sum_{l=1}^{r} \left(x_{j_1}^l - \bar{x}_{j_1}\right) \cdots \left(x_{j_\gamma}^l - \bar{x}_{j_\gamma}\right) w_l, \quad (5.2)$$

with

$$x^l = \begin{pmatrix} x_1^l \\ \vdots \\ x_n^l \end{pmatrix}, \forall l \in \left\{1, \ldots, r\right\}$$

and

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} = \operatorname{mean} x = \sum_{l=1}^{n} x^l w_l.$$

Lemma 10. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ be an operator such that*

$$f\left(x\right) = \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} f_\xi^{j_1,\ldots,j_\xi} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right), \quad (5.3)$$

where $f_\xi^{j_1,\ldots,j_\xi}$ real tensor for every $\xi \in \{1, \ldots, \gamma\}$,

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix},$$

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} = \int_{\mathbb{R}^n} xp\left(x\right) dx,$$

and $p(x)$ is a probability density function with mean $\bar{x}$.
Then, the mean $\bar{f}$ is

$$\bar{f} = \int_{\mathbb{R}^n} f(x)\, p(x)\, dx = \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} f_\xi^{j_1,\ldots,j_\xi} \mu_\xi^{j_1,\ldots,j_\xi}, \tag{5.4}$$

where $\mu_\xi^{j_1,\ldots,j_\xi}$ is the centred moment tensor of order ξ of $p(x)$.

Proof. By equation (5.3), the mean $\bar{f}$ can be written

$$\begin{aligned}\int_{\mathbb{R}^n} f(x)\, p(x)\, dx = \\ &= \int_{\mathbb{R}^n} \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} f_\xi^{j_1,\ldots,j_\xi} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right) p(x)\, dx \\ &= \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} f_\xi^{j_1,\ldots,j_\xi} \int_{\mathbb{R}^n} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right) p(x)\, dx\end{aligned}$$

and equation (5.4) is obtained by definition (5.1). □

Corollary 11. *Let $f, g : \mathbb{R}^n \longrightarrow \mathbb{R}$ be operators such that*

$$f(x) = \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} f_\xi^{j_1,\ldots,j_\xi} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right) \tag{5.5}$$

and

$$g(x) = \sum_{\xi=0}^{\gamma} \sum_{j_1,\ldots,j_\xi=1}^{n} g_\xi^{j_1,\ldots,j_\xi} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right), \tag{5.6}$$

where $f_\xi^{j_1,\ldots,j_\xi}$ and $g_\xi^{j_1,\ldots,j_\xi}$ real tensors for every $\xi \in \{1, \ldots, \gamma\}$,

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix},$$

$$\bar{x} = \begin{pmatrix} \bar{x}_1 \\ \vdots \\ \bar{x}_n \end{pmatrix} = \int_{\mathbb{R}^n} x p(x)\, dx,$$

and $p(x)$ *is a probability density function with mean* $\bar{x}$*.*
Then, the covariance between f *and* g *is*

$$\int_{\mathbb{R}^n} \left(f(x) - \bar{f}\right)(g(x) - \bar{g})\, p(x)\, dx =$$

$$= \sum_{\xi=0}^{2\gamma} \sum_{\xi'=0}^{\min\{\xi,\gamma\}} \sum_{j_1,\dots,j_\xi=1}^{n} f_{\xi'}^{j_1,\dots,j_{\xi'}} g_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi} \left(\mu_\xi^{j_1,\dots,j_\xi} - \mu_{\xi'}^{j_1,\dots,j_{\xi'}} \mu_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi}\right), \quad (5.7)$$

where $\mu_\xi^{j_1,\dots,j_\xi}$ *is the centred moment tensor of order* ξ *of* $p(x)$ *and* $\bar{f}$ *and* $\bar{g}$ *are the means of* f *and* g*, i.e.*

$$\bar{f} = \int_{\mathbb{R}^n} f(x)\, p(x)\, dx, \quad (5.8)$$

$$\bar{g} = \int_{\mathbb{R}^n} g(x)\, p(x)\, dx. \quad (5.9)$$

Proof. By equations (5.8) and (5.9), the covariance between f and g can be written

$$\int_{\mathbb{R}^n} \left(f(x) - \bar{f}\right)(g(x) - \bar{g})\, p(x)\, dx =$$

$$= \int_{\mathbb{R}^n} f(x)\, g(x)\, p(x)\, dx - \bar{f} \int_{\mathbb{R}^n} g(x)\, p(x)\, dx - \bar{g} \int_{\mathbb{R}^n} f(x)\, p(x)\, dx +$$

$$+ \bar{f}\bar{g} \int_{\mathbb{R}^n} p(x)\, dx \quad (5.10)$$

$$= \int_{\mathbb{R}^n} f(x)\, g(x)\, p(x)\, dx - \bar{f}\bar{g}.$$

By equations (5.5) and (5.6), it holds that

$$\int_{\mathbb{R}^n} f(x)\, g(x)\, p(x)\, dx =$$

$$= \int_{\mathbb{R}^n} \sum_{\xi=0}^{\gamma} \sum_{\xi'=0}^{\gamma} \sum_{j_1,\dots,j_\xi=1}^{n} \sum_{j'_1,\dots,j'_{\xi'}=1}^{n} f_\xi^{j_1,\dots,j_\xi} g_{\xi'}^{j'_1,\dots,j'_{\xi'}} \cdot$$

$$\cdot (x_{j_1} - \bar{x}_{j_1}) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right) \left(x_{j'_1} - \bar{x}_{j'_1}\right) \cdots \left(x_{j'_{\xi'}} - \bar{x}_{j'_{\xi'}}\right) p(x)\, dx$$

and, by changing indices and using definition (5.1),

$$\begin{aligned}
&\int_{\mathbb{R}^n} f\left(x\right) g\left(x\right) p\left(x\right) dx = \\
=&\sum_{\xi=0}^{2\gamma} \sum_{\xi'=0}^{\min\{\xi,\gamma\}} \sum_{j_1,\dots,j_\xi=1}^{n} f_{\xi'}^{j_1,\dots,j_{\xi'}} g_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi} \int_{\mathbb{R}^n} \left(x_{j_1} - \bar{x}_{j_1}\right) \cdots \left(x_{j_\xi} - \bar{x}_{j_\xi}\right) p\left(x\right) dx \\
=&\sum_{\xi=0}^{2\gamma} \sum_{\xi'=0}^{\min\{\xi,\gamma\}} \sum_{j_1,\dots,j_\xi=1}^{n} f_{\xi'}^{j_1,\dots,j_{\xi'}} g_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi} \mu_{\xi}^{j_1,\dots,j_\xi}.
\end{aligned} \tag{5.11}$$

Furthermore, by Lemma 10,

$$\begin{aligned}
\bar{f}\bar{g} =&\sum_{\xi=0}^{\gamma} \sum_{\xi'=0}^{\gamma} \sum_{j_1,\dots,j_\xi=1}^{n} \sum_{j'_1,\dots,j'_{\xi'}=1}^{n} f_{\xi}^{j_1,\dots,j_\xi} g_{\xi'}^{j'_1,\dots,j'_{\xi'}} \mu_{\xi}^{j_1,\dots,j_\xi} \mu_{\xi'}^{j'_1,\dots,j'_{\xi'}} \\
=&\sum_{\xi=0}^{2\gamma} \sum_{\xi'=0}^{\min\{\xi,\gamma\}} \sum_{j_1,\dots,j_\xi=1}^{n} f_{\xi'}^{j_1,\dots,j_{\xi'}} g_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi} \mu_{\xi'}^{j_1,\dots,j_{\xi'}} \mu_{\xi-\xi'}^{j_{\xi'+1},\dots,j_\xi},
\end{aligned} \tag{5.12}$$

where the last equality has been obtained by rearranging the indices.
Finally, equation (5.7) can easily be obtained by equations (5.10), (5.11) and (5.12). □

Theorem 12. *Let $f : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ be a polynomial operator of degree γ and let $x, x' \in \mathbb{R}^n$ be two random variable with same mean and centred moment tensors up to the γ-th order.*
Then, the random variables $f\left(x\right)$ and $f\left(x'\right)$ have the same mean.
Furthermore, if x, x' have the same centred moment tensors up to the 2γ-th order, then $f\left(x\right)$ and $f\left(x'\right)$ have the same covariance matrix.

Proof. It comes easily from Lemma 10 and Corollary 11. □

Theorem 12 proves that, approximating a probability density function $p\left(x\right)$ with a discrete weighted ensemble having same mean and centred moment tensors up to a certain order γ is enough to obtain the exact mean of the γ-th order Taylor expansion of an operator f applied to x (or the exact covariance of the $\frac{\gamma}{2}$-th order Taylor expansion of f).
This explains why second order exact sampling performs better than a simple linearisation and, on the other hand, this is a motivation to further exploit higher moments.
A general way to sample an ensemble with given mean and moments is to

solve a non-linear system of equations. Looking at equation (4.24), the C matrix can be intended as a change of variable leading to a standardised normal distribution with covariance I_r:

$$A^b_{i-1} = C I_r C^T$$

and the procedure used for the second order exact sampling is a way to sample points with zero mean and same second order centred moment (i.e. covariance) of this standardised Gaussian. In fact, equations from (4.25) to (4.31) imply

$$\left(U^T V^{-1}\right) w = 0$$

and

$$\left(U^T V^{-1}\right) W \left(U^T V^{-1}\right)^T = I_r.$$

In order to guarantee higher order convergence, it is sufficient to impose centred moment equivalence conditions (say up to order γ) using equation (5.2)

$$\forall \xi \in \{2, \ldots, \gamma\}, j_1, \ldots, j_\xi \in \{1, \ldots, r\}, \sum_{l=1}^{\tilde{r}} z^l_{j_1} \cdots z^l_{j_\xi} w_l = \mu_\xi^{j_1,\ldots,j_\xi},$$

where

$$z^l_j = u_{l,j} {v_l}^{-1}$$

and $u_{l,j}$ is the element of U in the l-th row and j-th column.
All together, the non-linear system to be solved is

$$\begin{cases} v^T v = 1, \\ U^T v = 0, \\ \forall \xi \in \{2, \ldots, \gamma\}, j_1, \ldots, j_\xi \in \{1, \ldots, r\}, \displaystyle\sum_{l=1}^{\tilde{r}} u_{l,j_1} \cdots u_{l,j_\xi} v_l^{2-\xi} = \mu_\xi^{j_1,\ldots,j_\xi}, \end{cases} \tag{5.13}$$

where the centred moment tensors value are defined by the following lemma.

Lemma 13. *Let $x \in \mathbb{R}^n, k \in \mathbb{N}^n$ be a vector and a multi-index such that*

$$x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}$$

and

$$k = \begin{pmatrix} k_1 \\ \vdots \\ k_n \end{pmatrix}.$$

Then

$$\int_{\mathbb{R}^n} {x_1}^{k_1} \cdots {x_n}^{k_n} \mathcal{N}\left(x; 0, I_n\right) dx =$$

$$= \begin{cases} 0, & \textit{if any index in } k \textit{ is an odd number;} \\ \dfrac{k_1!}{\sqrt{2}^{k_1}\left(\frac{k_1}{2}!\right)} \cdots \dfrac{k_n!}{\sqrt{2}^{k_n}\left(\frac{k_n}{2}!\right)}, & \textit{otherwise.} \end{cases}$$

Proof. Since, splitting the Gaussian in its unidimensional components,

$$\int_{\mathbb{R}^n} {x_1}^{k_1} \cdots {x_n}^{k_n} \mathcal{N}\left(x; 0, I_n\right) dx = \prod_{i=1}^{n} \int_{\mathbb{R}^n} {x_i}^{k_i} \mathcal{N}\left(x_i; 0, 1\right) dx_i,$$

it is sufficient to prove the Lemma in one dimension.
Let $k \in \mathbb{N}$ be a non negative integer and $f_k : \mathbb{R} \longrightarrow \mathbb{R}$ function such that

$$f_k\left(x\right) = \frac{1}{\sqrt{2\pi}} x^k e^{-\frac{1}{2}x^2}.$$

If k is an odd number, f_k is an odd function, i.e.

$$f_k\left(-x\right) = -f_k\left(x\right), \forall x \in \mathbb{R},$$

and its integral is zero.
If $k = 0$, then f_0 is the Gaussian and integrates to 1.
Finally, if $k = 2\alpha$ is an even positive number, then, integrating by parts,

$$\int_{\mathbb{R}} f_{2\alpha}\left(x\right) dx = \int_{\mathbb{R}} x^{2\alpha-1} \cdot \frac{1}{\sqrt{2\pi}} x e^{-\frac{1}{2}x^2} dx = \left(2\alpha - 1\right) \int_{\mathbb{R}} f_{2(\alpha-1)}\left(x\right) dx,$$

and, by induction over α,

$$\int_{\mathbb{R}} f_{2\alpha}\left(x\right) dx = 1 \cdot 3 \cdot \ldots \cdot \left(2\alpha - 1\right) = \frac{(2\alpha)!}{2^\alpha\left(\alpha!\right)}.$$

□

However, a trick can be used to reduce the complexity of system (5.13). Since the mean and all the odd-order moment tensors of a symmetric distribution (as a zero-mean-Gaussian is) are zero (it can be easily proved as done in Lemma 13), it is sufficient to impose the equivalence on the even moments and then symmetrise the ensemble. Furthermore, adding $0 \in \mathbb{R}^r$ as ensemble member, it is not necessary to care about the weights summing to one, as all

the missing weight can be given to the 0 member.
Then, solving the system

$$\begin{cases} \forall \xi \in \{1, \ldots, \gamma\}, j_1, \ldots, j_{2\xi} \in \{1, \ldots, r\}, \sum_{l=1}^{\tilde{r}} u_{l,j_1} \cdots u_{l,j_{2\xi}} v_l^{2-2\xi} = \mu_{2\xi}^{j_1,\ldots,j_{2\xi}}, \end{cases} \tag{5.14}$$

and then using $\tilde{U}$ and $\tilde{v}$ in place of U and v, such that

$$\tilde{U} = \left(\begin{array}{ccc} & \frac{1}{\sqrt{2}} U & \\ \hline & -\frac{1}{\sqrt{2}} U & \\ \hline 0 & \cdots & 0 \end{array} \right) \tag{5.15}$$

and

$$\tilde{v} = \left(\begin{array}{c} \frac{1}{\sqrt{2}} v \\ \hline \frac{1}{\sqrt{2}} v \\ \hline \sqrt{1 - v^T v} \end{array} \right), \tag{5.16}$$

guarantee zero mean and equivalent moment tensors up to order $2\gamma + 1$.
Given the higher number of constrains, the ensemble size $\tilde{r}$ will be higher then the $r + 1$ size used in SEIK. This is an unpleasant effect, since the computational cost of the method is directly connected with the number of ensemble members.
To avoid this problem, it will be now presented a strategy to sample an ensemble with high order in the most relevant PCA directions of the background probability distribution and with smaller order (but not smaller then two) in the less relevant ones.
We called this procedure "minimum decreasing $(2\gamma + 1)$-th order exact sampling".

First of all, let's choose r even, such that

$$r = 2\alpha.$$

Then, a PCA decomposition of P^b_{i-1} is needed. Since, by equations (4.23) and (4.24),

$$P^b_{i-1} = (L_{i-1}C)\,(L_{i-1}C)^T\,,$$

the fastest way is to use a Singular Value Decomposition algorithm (see [17] for an example) on the $(L_{i-1}C)$ matrix, obtaining the decomposition

$$L_{i-1}C = \Sigma\Lambda\Omega_r,$$

where Σ is a $N \times r$ matrix with orthonormal columns, Λ is a diagonal $r \times r$ matrix with decreasing positive eigenvalues on the diagonal and Ω_r is a $r \times r$ orthogonal matrix.
Thus, P^b_{i-1} can be written

$$P^b_{i-1} = \tilde{L}_{i-1} I_r \tilde{L}^T_{i-1}, \tag{5.17}$$

where

$$\tilde{L}_{i-1} = \Sigma\Lambda.$$

Decomposition (5.17) says which change of base is needed to go from the standardized Gaussian covariance matrix I_r to P^b_{i-1}, transforming it along orthogonal direction and starting from the most important ones. Then, the identity matrix can be substituted by the usual minimum second order exact sampling decomposition

$$I_r = U^T V^{-1} W V^{-1} U \tag{5.18}$$

built in the following way:

1. first, a solution of system (5.14) is computed once and for all and stocked into U^0 and v^0 using β and α in place of r and $\tilde{r}$ respectively, such that β is the maximum integer for which system (5.14) has a solution (this is to maximize the number of dimensions approximated with the highest order sampling).

2. When decomposition (5.18) is needed, U^0 is multiplied for a random orthogonal matrix Ω_β to add stochasticity (actually applying a randomly rotation and/or symmetry to the ensemble) and then completed to a $\alpha \times \alpha$ orthogonal matrix Ω_α,

$$\Omega_\alpha = \left(\begin{array}{c|c} U^0\Omega_\beta & \cdots \end{array} \right),$$

3. $\tilde{U}$ and v are built using the trick above, equations (5.15) and (5.16), i.e.

$$\tilde{U} := \begin{pmatrix} \frac{1}{\sqrt{2}}\Omega_\alpha \\ \hline -\frac{1}{\sqrt{2}}\Omega_\alpha \\ \hline 0 \quad \cdots \quad 0 \end{pmatrix} \tag{5.19}$$

and

$$v := \begin{pmatrix} \frac{1}{\sqrt{2}}v^0 \\ \hline \frac{1}{\sqrt{2}}v^0 \\ \hline \sqrt{1 - v^{0T}v^0} \end{pmatrix}. \tag{5.20}$$

Note that v is a normalized $(r+1)$-dimensional vector and that $\tilde{U}$ has dimensions $(r+1) \times \alpha$.

4. Finally, $\tilde{U}$ is completed by minimum second order exact sampling procedure to obtain the $(r+1) \times r$ matrix U, namely

$$U = \left(\begin{array}{c|c} \tilde{U} & \cdots \end{array} \right),$$

$$U^T v = 0$$

and

$$U^T U = I_r.$$

Thus, the ensemble in the columns of $\left(U^T V^{-1}\right)$, with weights w (obtained by the squares of the coordinates of v as usual), is second order exact, but, if projected in its first α coordinates, it is third order exact, while it is $(2\gamma+1)$-th order exact in its first β coordinates.

Note that if an odd $r = 2\alpha + 1$ is needed, then it is sufficient to modify equation (5.19) adding another line of zeros at the bottom, and equation (5.20) such that the last two coordinates of v squared sum to $1 - {v^0}^T v^0$.

5.2 Accounting for the model error

Usually, working with the model error covariance matrix Q is difficult, because quantifying it is quite problematic, and (when not directly neglected) inflation strategies are preferred. These are methods to increase the covariance matrix of an ensemble in order to take partially into account the model error.

For example, Pham (in [37] and [38]) propose to substitute equation (4.41), i.e.

$$A_i^f := \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} + \left(T^T W^{-1} T\right)^{-1}.$$

with a simpler equation, namely

$$A_i^f := \rho \left(T^T W^{-1} T\right)^{-1},$$

where $\rho > 1$ is a positive number that increases the uncertainty in a constant multiplicative way.

However, some processes (as in marine biogeochemistry) have not negligible errors that should be considered more accurately. Thus, in this section, a novel strategy to take into account Q matrix effects is presented.

Looking back to approximation (4.14),

$$Q_i \approx L_i \left(L_i^T L_i\right)^{-1} L_i^T Q_i L_i \left(L_i^T L_i\right)^{-1} L_i^T,$$

it was obtained by an orthogonal projection into the state error subspace spanned by the columns of L_i. This expression for Q can be very far from reality, since there are no reasons for the model error to be mainly included in such subspace. Then, instead of projecting the error, the main idea is to split it into two independent components, one along the columns of L_i and one everywhere else. The former is used to correct the forecast covariance A_i^f, since it affects the error subspace. The latter is treated as a noise and accounted as added observation error.

To achieve this result, it is useful look at Theorem 8: if P is chosen equal to the inverse of A, the two variable x and y become independent. The following lemma captures the point.

Lemma 14. *Let L and L' be matrices of dimensions $n \times m$ and $n \times l$ respectively. Furthermore, let A be symmetric positive-definite $n \times n$ matrices such that*

$$L^T A^{-1} L' = 0. \tag{5.21}$$

Then

$$\begin{aligned}\mathcal{N}\left(Lx + L'y; La + L'b, A\right) &= \\ &= \mathcal{N}\left(x; a, \left(L^T A^{-1} L\right)^{-1}\right) \mathcal{N}\left(y; b, \left(L'^T A^{-1} L'\right)^{-1}\right),\end{aligned} \tag{5.22}$$

where $x, a \in \mathbb{R}^m, y, b \in \mathbb{R}^l$.

Proof. This can be proved by Theorem 8, but here a more direct approach is preferred.
In fact, by hypothesis (5.21),

$$\begin{aligned}\left(Lx + L'y - (La + L'b)\right)^T A^{-1} \left(Lx + L'y - (La + L'b)\right) &= \\ &= \left(L(x-a) + L'(y-b)\right)^T A^{-1} \left(L(x-a) + L'(y-b)\right) \\ &= (x-a) L^T A^{-1} L (x-a) + (y-b) L'^T A^{-1} L' (y-b)\end{aligned}$$

and equation (5.22) follows by applying exponentials and normalisation. □

In the following part, Lemma 14 is applied starting from the SEEK's equations first, because they are easier and useful to understand the method. The exact sampling strategy is later adopted to obtain a SEIK-like counterpart, with all the benefits of higher order approximation.
Thus, from SEEK's equation (4.3), namely

$$P_i^f = Q_i + L_i A_{i-1}^b L_i^T, \tag{5.23}$$

it is possible to complete the columns of L_i to a base of $\mathbb{R}^N$, storing the missing vectors in the columns of the $N \times (N-r)$ matrix L', such that

$$L_i^T Q_i^{-1} L' = 0,$$

and this can always be done thanks to Gram-Schmidt orthonormalisation algorithm (obviously, Q_i is considered full rank).
Now, by Lemma 14,

$$Q_i = \left(\begin{array}{c|c} L_i & L' \end{array}\right) \left(\begin{array}{c|c} \left(L_i^T Q_i^{-1} L_i\right)^{-1} & 0 \\ \hline 0 & \left(L'^T Q_i^{-1} L'\right)^{-1} \end{array}\right) \left(\begin{array}{c} L_i^T \\ \hline L'^T \end{array}\right)$$

and then

$$Q_i = L_i Q_i^L L_i^T + L' Q^{L'} L'^T, \tag{5.24}$$

where

$$Q_i^L = \left(L_i^T Q_i^{-1} L_i\right)^{-1}$$

and

$$Q^{L'} = \left(L'^T Q_i^{-1} L'\right)^{-1}.$$

The matrix Q_i^L represent the covariance of the normal distribution obtained by sectioning the model error Gaussian along the r-dimensional subspace generated by the columns of L_i. On the other hand, $Q^{L'}$ is the covariance matrix of the residual model error out of that subspace.
Substituting Q_i in equation (5.23) with equation (5.24), we obtain

$$P_i^f = L_i Q_i^L L_i^T + L' Q^{L'} L'^T + L_i A_{i-1}^b L_i^T = L' Q^{L'} L'^T + L_i A_i^f L_i^T, \tag{5.25}$$

where

$$A_i^f \coloneqq A_{i-1}^b + Q_i^L. \tag{5.26}$$

Now, to understand what is happening, it is useful to answer the question "what is the probability to measure y_i, given $p_i^f(x_i)$?".
Let's call such probability density function $p_i^l(y_i)$, or likelihood probability. By the law of total probability (Theorem 3) and definition (3.2),

$$p_i^l(y_i) = \int_{\mathbb{R}^N} p(y_i|x_i)\, p_i^f(x_i) = \int_{\mathbb{R}^N} p_{\mathcal{H}_i(x_i)}(y_i)\, p_i^f(x_i)$$

In the Extended Kalman-Filter (and SEEK) formalism, this equation becomes, integrating equation (3.41),

$$p_i^l(y_i) = \mathcal{N}\left(y_i - h_i\left(x_i^f\right); 0, P_i^l\right) = \mathcal{N}\left(y_i; h_i\left(x_i^f\right), P_i^l\right), \tag{5.27}$$

where

$$P_i^l \coloneqq R_i + H_i P_i^f H_i^T.$$

The last term in the definition of P_i^l represent the covariance of the observation operator (here linearised) applied to the forecast probability. Replacing the forecast covariance P_i^f by equation (5.25), we have

$$P_i^l = R_i^l + H_i L_i A_i^f \left(H_i L_i\right)^T, \tag{5.28}$$

where

$$R_i^l \coloneqq R_i + H_i L' Q^{L'} L'^T H_i^T. \tag{5.29}$$

Thus, R_i^l represents an amplification of the observation error covariance R_i, caused by the Gaussian noise with covariance $Q^{L'}$ along the unmonitored (that is to say neglected by the SEEK's dimensionality-reducing strategy) subspace spanned by the columns of L'.
This amplification effect can be very strong, in particular when measurements have low uncertainty and model error is relevant. Not taking it into account leads to an overestimation of the precision of the observation operator and, subsequently, to an underestimation of the analysis covariance.
Since L' dimensions, $N \times (N-r)$, can be huge, it is desirable to avoid the direct calculation. Then, using equation (5.24) in (5.29),

$$R_i^l = R_i + R_i^Q - H_i L_i Q_i^L \left(H_i L_i\right)^T , \tag{5.30}$$

where

$$R_i^Q \coloneqq H_i Q_i H_i^T. \tag{5.31}$$

The R_i^Q matrix has dimensions $n \times n$ but, in the most general case, the computational cost of its calculation can be very heavy. However, in most cases, both Q_i and H_i usually have a quite simple form and the matrix product in equation (5.31) can be highly simplified.
A very simple but common example is when the observation operator represents some measurements of some system variables in some places. In this case, H_i is mainly composed by zeros with some sparse ones and Q_i^H is simply a sub-matrix of Q_i.
Summarizing, to have a SEEK-like algorithm taking into account the noise effect induced by the model error, it is sufficient to use expression (5.26) for A_i^f and substitute R_i with the corrected matrix R_i^l.
Now, to add the exact sampling strategy we proceed as follows.
Starting from SEIK's forecast covariance equation (4.40), i.e.

$$P_i^f = Q_i + L_i \left(T^T W^{-1} T\right)^{-1} L_i^T ,$$

and substituting Q_i via equation (5.24), we have

$$P_i^f = L_i Q_i^L L_i^T + L' Q^{L'} L'^T + L_i \left(T^T W^{-1} T\right)^{-1} L_i^T = L' Q^{L'} L'^T + L_i A_i^f L_i^T ,$$

where

$$A_i^f \coloneqq \left(T^T W^{-1} T\right)^{-1} + Q_i^L.$$

By Sylvester's Theorem, Q_i^L, $Q^{L'}$ and A_i^f can be factorised as

$$Q_i^L = C C^T, \tag{5.32}$$

$$Q^{L'} = C' C'^T \tag{5.33}$$

and

$$A_i^f = C''C''^T, \tag{5.34}$$

where C, C' and C'' are matrices of dimensions $r \times r$, $(N-r)\times(N-r)$ and $r \times r$ respectively.
Furthermore, using a Singular Value Decomposition,

$$L_i C'' = \Sigma\Lambda\Omega_r, \tag{5.35}$$

where Σ is a $N \times r$ matrix with orthonormal columns, Λ is a diagonal $r \times r$ matrix with decreasing positive eigenvalues on the diagonal and Ω_r is a $r \times r$ orthogonal matrix.
Thus, by equations (5.33) and (5.35), P_i^f can be written

$$P_i^f = L'C'\left(L'C'\right)^T + \tilde{L}_i\tilde{L}_i^T, \tag{5.36}$$

where

$$\tilde{L}_{i-1} = \Sigma\Lambda. \tag{5.37}$$

Equation (5.36) is equivalent to

$$P_i^f = \left(\begin{array}{c|c} \tilde{L}_i & L'C' \end{array}\right)\left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & I_{N-r} \end{array}\right)\left(\begin{array}{c} \tilde{L}_i^T \\ \hline \left(L'C'\right)^T \end{array}\right) \tag{5.38}$$

and the big matrix in the middle of the right-hand side is the identity matrix of dimension N.
In order to apply the exact sampling technique, we need to decompose such matrix but, since we want to preserve the blocks structure, we prefer to work separately on I_r and I_{N-r}. So we look for factorisations

$$I_r = U^T V^{-1} W V^{-1} U \tag{5.39}$$

and

$$I_{N-r} = U'^T V'^{-1} W' V'^{-1} U',$$

such that

$$I_N = \left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & I_{N-r} \end{array}\right) =$$

$$= \left(\begin{array}{c|c} U & 0 \\ \hline 0 & U' \end{array}\right)^T \left(\begin{array}{c|c} V & 0 \\ \hline 0 & V' \end{array}\right)^{-1} \left(\begin{array}{c|c} W & 0 \\ \hline 0 & W' \end{array}\right) \left(\begin{array}{c|c} V & 0 \\ \hline 0 & V' \end{array}\right)^{-1} \left(\begin{array}{c|c} U & 0 \\ \hline 0 & U' \end{array}\right) \tag{5.40}$$

is an exact sampling factorisation.

Note that such factorisation cannot lead to an exact sampling of order higher then three. In fact, due to the block-structure of the first (or last) matrix in the right hand side of equation (5.40), any order centred moment tensor obtained by the ensemble built by this factorisation has zero elements at positions with indices that are not all smaller then (or equal to) r or not all bigger then r. Thus, it is impossible to match all the non zero Gaussian's forth order moment tensor values, that, by Lemma 13, have the following property

$$\forall j_1, j_2 \in \{1, \dots, N\}, \mu_4^{j_1,j_1,j_2,j_2} = \begin{cases} 3, & \text{if } j_1 = j_2; \\ 1, & \text{otherwise.} \end{cases}$$

Then, the idea is to build U (and the corresponding weights) using a modified version of the minimum decreasing high order exact sampling algorithm exposed in Section 5.1 to have a high order exact sampling on the principal components identified by the columns of $\tilde{L}_i$.

U' is built by a third order exact sampling instead.

Entering into details, if $r = 2\alpha - 1$ is an odd number, the algorithm steps are:

1. Calculate U^0, v^0, β and Ω_α as in the first two points of the minimum decreasing high order exact sampling algorithm in Section 5.1. Note that U^0, v^0 and β are already available as needed to sample the previous ensemble.

2. Differently from the third step of the algorithm in Section 5.1, $\tilde{U}$ and v are defined as

$$\tilde{U} := \left(\begin{array}{c} \frac{1}{\sqrt{2}}\Omega_\alpha \\ \hline -\frac{1}{\sqrt{2}}\Omega_\alpha \end{array} \right)$$

and

$$v := \left(\begin{array}{c} \frac{1}{\sqrt{2}}v^0 \\ \hline \frac{1}{\sqrt{2}}v^0 \end{array} \right),$$

with v a $(r+1)$-dimensional vector and $\tilde{U}$ matrix with dimensions $(r+1) \times \alpha$.

3. The $(r+1) \times r$ matrix U is obtained completing $\tilde{U}$ by minimum second order exact sampling procedure, namely

$$U = \left(\begin{array}{c|c} \tilde{U} & \cdots \end{array} \right),$$

$$U^T v = 0$$

and

$$U^T U = I_r.$$

4. U' is any $\tilde{r}' \times (N - r)$ third order exact sampling matrix, with square rooted weights vector $\tilde{v}' \in \mathbb{R}^{\tilde{r}'}$, where $\tilde{r}'$ is a real number. An easy way

to build them, with $\tilde{r}' = 2\,(N-r)$, is

$$U' = \left(\begin{array}{c} \frac{1}{\sqrt{2}}\Omega_{N-r} \\ \hline -\frac{1}{\sqrt{2}}\Omega_{N-r} \end{array}\right)$$

and

$$\tilde{v}' = \frac{1}{2\,(N-r)}\mathbf{1}_{2(N-r)\times 1},$$

where Ω_{N-r} is a random orthogonal $(N-r)\times(N-r)$ matrix.

5. Finally,

$$v' := \sqrt{1 - v^T v}\,\tilde{v}'.$$

If $r = 2\alpha$ is an even number, it is sufficient to add a zero row in the bottom of $\tilde{U}$ and a random positive number in the bottom of v such that $v^T v < 1$. Now, using this definitions in equation (5.40), the ensemble obtained is second order exact, increasing to third order everywhere but on the less significant components stored in the columns of $\tilde{L}_i$, and reaching order $(2\gamma + 1)$ on the most relevant β components.
Looking at equations (5.38) and (5.40), such ensemble is built, following the usual procedure, by summing the mean x_i^f to each column of the matrix

$$\left(\begin{array}{c|c} \tilde{L}_i & L'C' \end{array}\right) \left(\begin{array}{c|c} U & 0 \\ \hline 0 & U' \end{array}\right)^T \left(\begin{array}{c|c} V & 0 \\ \hline 0 & V' \end{array}\right)^{-1},$$

which is equivalent to sum x_i^f to the columns of both the matrices

$$\tilde{L}_i U^T V^{-1}$$

and

$$L'C'U'^T V'^{-1}.$$

Thus, after applying h_i to each ensemble member, the n-dimensional vectors obtained are stocked into the columns of the matrices Y_i and Y_i' respectively,

with corresponding weights w and w', obtained by squaring the coordinates of v and v'.
Then, the mean y_i^l of p_i^l can be calculated as

$$y_i^l = Y_i w + Y_i' w', \tag{5.41}$$

while the covariance expression is

$$P_i^l = R_i + Y_i W Y_i^T + Y_i' W' Y_i'^T - y_i^l y_i^{lT}, \tag{5.42}$$

where W and W' are the usual weight diagonal matrices. Now, note that U and v (and consequently V and W) do not lead to an exact sampling decomposition of I_r because, even if equation (5.39) holds, v is not normalised, falling to match the centred moment tensor of order zero. On the other hand, it is sufficient to add a zero row at the bottom of U and the root squared missing weight at the bottom of v, i.e.

$$\hat{U} := \left(\begin{array}{ccc} & U & \\ \hline 0 & \cdots & 0 \end{array} \right)$$

and

$$\hat{v} := \left(\begin{array}{c} v \\ \hline \sqrt{1 - v^T v} \end{array} \right),$$

to have a good exact sampling decomposition

$$I_r = \hat{U}^T \hat{V}^{-1} \hat{W} \hat{V}^{-1} \hat{U},$$

with $\hat{V}$ and $\hat{W}$ obtained from $\hat{v}$ as usual.
The added ensemble member, corresponding to the zero line, is the mean x_i^f, that, transformed by h_i and added as column of Y_i, forms the ensemble $\hat{Y}_i$ with weights $\hat{w}$, that is to say

$$\hat{Y}_i := \left(\begin{array}{c|c} Y_i & y_i^0 \end{array} \right)$$

and

$$\hat{w} := \left(\begin{array}{c|c} w & \psi \end{array} \right),$$

where

$$y_i^0 := h_i\left(x_i^f\right)$$

and

$$\psi := 1 - v^T v.$$

Thus, the mean and covariance of the ensemble $\hat{Y}_i$ are

$$y_i^f := \hat{Y}_i \hat{w} = Y_i w + \psi y_i^0 \tag{5.43}$$

and

$$R_i^f := \hat{Y}_i \hat{W} \hat{Y}_i^T - y_i^f {y_i^f}^T = Y_i W Y_i^T + \psi y_i^0 {y_i^0}^T - y_i^f {y_i^f}^T \tag{5.44}$$

respectively. By equations (5.41) and (5.43)

$$y_i^l = y_i^f + y_i', \tag{5.45}$$

where

$$y_i' = Y_i' w' - \psi y_i^0,$$

and, using equation (5.45) in (5.42)

$$P_i^l = R_i + Y_i W Y_i^T + Y_i' W' {Y_i'}^T - y_i^f {y_i^f}^T - y_i' {y_i'}^T - y_i^f {y_i'}^T - y_i' {y_i^f}^T. \tag{5.46}$$

Finally, by equation (5.44), equation (5.46) becomes

$$P_i^l = R_i^l + R_i^f, \tag{5.47}$$

where

$$R_i^l := R_i + R' - \left(y_i^f {y_i'}^T + y_i' {y_i^f}^T\right) \tag{5.48}$$

and

$$R' := Y_i' W' {Y_i'}^T - \psi y_i^0 {y_i^0}^T - y_i' {y_i'}^T.$$

Thanks to Theorem 12, y_i^f and R_i^f do not depend (up to a certain order) by the choice of the exact sample ensemble. Then, $\hat{Y}_i$ and $\hat{w}$ can be replaced by any other (preferably minimum) ensemble matrix and weights.

Calling Y_i^f such ensemble, with corresponding matrices and vectors U^f, V^f, W^f, v^f and w^f, then R_i^f can be written

$$R_i^f = Y_i^f T \left(T^T {W^f}^{-1} T\right)^{-1} \left(Y_i^f T\right)^T,$$

where T as in SEIK's equation (4.39).

Finally, equation (5.47) becomes

$$P_i^l = R_i^l + Y_i^f T \left(T^T {W^f}^{-1} T\right)^{-1} \left(Y_i^f T\right)^T, \tag{5.49}$$

where $\left(Y_i^f T\right)$ represents the action of h_i as in the SEIK algorithm, while $\left(T^T {W^f}^{-1} T\right)^{-1}$ is the covariance matrix A_i^f after a convenient change of base, in fact, by equations (5.34), (5.35) and (5.37),

$$L_i A_i^f L_i^T = L_i C'' C''^T L_i^T = \tilde{L}_i \tilde{L}_i^T = L_i^a \left(T^T {W^f}^{-1} T\right)^{-1} L_i^{aT},$$

with

$$L_i^a = \tilde{L}_i U^f {V^f}^{-1} T.$$

Equation (5.49) is the SEIK-like counterpart of equation (5.28), but the use of the exact sampling method brought some improvements.
In fact, equation (5.45) says that the mean y_i^f calculated in the r-dimensional reduced error subspace must be corrected by y_i' in order to better approximate y_i^l.
Moreover, the correction R_i^l to R_i appears more articulated than in SEEK-like equations: the first two terms at the right hand side of equation (5.48) is the exact sampling counterpart of equation (5.29), with R' representing the action of the function $h_i\left(x_i\right) - h_i^0$ on the noise part of the model error; the last term instead is a further correction that takes into account the covariance generated by h_i between the two independent parts in which equation (5.24) splits the model error.
However, as happened in equation (5.29), the primed elements in equations (5.45) and (5.48) are often too heavy to calculate directly, since Y' can have huge dimensions.
Then, the idea is to follow a strategy similar to the one used for equation (5.30) but again taking advantage from the exact sampling technique in the following way.
By equations (5.24), (5.32) and (5.33),

$$Q_i = L_i C C^T L_i^T + L' C' C'^T L'^T,$$

that is equivalent to

$$Q_i = \left(\begin{array}{c|c} L_i C & L'C' \end{array} \right) \left(\begin{array}{c|c} I_r & 0 \\ \hline 0 & I_{N-r} \end{array} \right) \left(\begin{array}{c} \left(L_i C\right)^T \\ \hline \left(L'C'\right)^T \end{array} \right). \tag{5.50}$$

Equation (5.50) has the same form of equation (5.38), with the only difference of $L_i C$ in place of $\tilde{L}_i$. Thus, any calculation done up to equation (5.48) can be repeated in order to obtain (with third order agreement) the mean y_i^Q and covariance R_i^Q of $h_i(x_i)$, with x_i following the distribution $\mathcal{N}\left(x_i; x_i^f, Q_i\right)$ instead of $p_i^f(x_i) = \mathcal{N}\left(x_i; x_i^f, P_i^f\right)$.

Then, as in equations (5.45), (5.47) and (5.48),

$$y_i^Q = \hat{y}_i^f + y_i', \tag{5.51}$$

$$R_i^Q = \hat{R}_i^f + R' - \left(\hat{y}_i^f y_i'^T + y_i'\hat{y}_i^{fT}\right), \tag{5.52}$$

where the primed element has been left unchanged, while $\hat{y}_i^f$ and $\hat{R}_i^f$, substituting y_i^f and R_i^f, are the mean and covariance of $h_i(x_i)$, with x_i following the distribution $\mathcal{N}\left(x_i; x_i^f, L_iC\left(L_iC\right)^T\right)$ instead of $\mathcal{N}\left(x_i; x_i^f, \tilde{L}_i\tilde{L}_i^T\right)$.

Since, by Theorem 12, y_i^Q and R_i^Q are not dependent from the ensemble (up to a certain order), smart ensembles can be chosen in order to simplify the calculation. There is not a known general not expensive way to compute them, but in most cases, the simple form of Q_i and h_i can be exploited in order to obtain a fast result (a real example is provided in details in Section 6.4).

Thus, by equations (5.51) and (5.52),

$$y_i' = y_i^Q - \hat{y}_i^f,$$

$$R' = R_i^Q - \hat{R}_i^f + \hat{y}_i^f y_i'^T + y_i'\hat{y}_i^{fT},$$

and the primed elements are obtained without using large dimensions matrices, like Y'.

All together, the obtained filter can be summarized as follows (some names have been changed a little in order to have a clearer summary):

- U^0 and v^0 (and consequently v, w, V, W and T) are prepared as described in the minimum descending γ-th order exact sampling algorithm in Section 5.1, page 53 and equation (4.39).

- At step i, x_{i-1}^b and P_{i-1}^b, in decomposed form

  $$P_{i-1}^b = L_{i-1}A_{i-1}^b {L_{i-1}}^T,$$

  are received from the previous step, along with the observation y_i and its covariance R_i at time t_i.

- C^b is computed (with an SVD or Cholesky decomposition) such that

  $$A_{i-1}^b = C^b {C^b}^T,$$

the SVD decomposition

$$L_{i-1}C^b = \Sigma^b \Lambda^b {\Omega_r^b}^T$$

is prepared, and U^b is randomly generated as U in the minimum descending γ-th order exact sampling algorithm.

- The ensemble in the columns of X^b, i.e.

$$X^b = L_{i-1}C^b \Omega_r^b {U^b}^T V^{-1} + x_{i-1}^b \mathbf{1}_{1\times(r+1)}, \tag{5.53}$$

  is evolved via m_i, leading to X^f.

- The forecast mean, (reduced) covariance and its base are computed as follows

$$x_i^f = X^f w,$$
$$A_i^f = \left(T^T W^{-1} T\right)^{-1} + \left({L_i}^T Q_i^{-1} L_i\right)^{-1},$$

  and

$$L_i = X^f T.$$

- C^f and $\hat{C}^f$ are computed (with an SVD or Cholesky decomposition) such that

$$A_i^f = C^f {C^f}^T$$

  and

$$\left({L_i}^T Q_i^{-1} L_i\right)^{-1} = \hat{C}^f \hat{C}^{fT},$$

  the SVD decompositions

$$L_i C^f = \Sigma^f \Lambda^f {\Omega_r^f}^T$$

  and

$$L_i \hat{C}^f = \hat{\Sigma}^f \hat{\Lambda}^f \hat{\Omega}_r^{fT}$$

  are prepared, and U^f and $\hat{U}^Q$ are randomly generated as U in the minimum descending γ-th order exact sampling algorithm.

- The operator h_i is applied to the ensembles in the columns of $\tilde{X}^f$ and $\hat{X}^f$, i.e.

$$\tilde{X}^f = L_i C^f \Omega_r^f {U^f}^T V^{-1} + x_i^f \mathbf{1}_{1\times(r+1)}$$

  and

$$\hat{X}^f = L_i \hat{C}^f \hat{\Omega}_r^f \hat{U}^{fT} V^{-1} + x_i^f \mathbf{1}_{1\times(r+1)},$$

  obtaining Y^f and $\hat{Y}^f$ respectively.

- The mean y_i^Q and covariance R_i^Q of $h_i\left(x_i\right)$, with x_i following the distribution $\mathcal{N}\left(x_i; x_i^f, Q_i\right)$, are computed with some *ad hoc* method.

- The likelihood mean and observation covariance are calculated as

$$y_i^l = y_i^f + y_i'$$

and

$$R_i^l = R_i + R' - \left(y_i^f {y_i'}^T + y_i' {y_i^f}^T\right), \tag{5.54}$$

where

$$\begin{aligned} y_i^f &= Y^f w, \\ y_i' &= y_i^Q - \hat{y}_i^f, \\ R' &= R_i^Q - \hat{R}_i^f + \hat{y}_i^f {y_i'}^T + y_i' \hat{y}_i^{fT}, \\ \hat{y}_i^f &= \hat{Y}^f w \end{aligned}$$

and

$$\hat{R}_i^f = \hat{Y}^f W \hat{Y}^{fT} - \hat{y}_i^f \hat{y}_i^{fT}.$$

- Finally, the analysis mean and covariance (in decomposed form) are computed as follows

$$x_i^a = x_i^f + L_i^a \tilde{A}_i^a \left(Y^f T\right)^T {R_i^l}^{-1} \left(y_i - y_i^l\right)$$

and

$$P_i^a = L_i A_i^a {L_i}^T, \tag{5.55}$$

where

$$\begin{aligned} A_i^a &= \Gamma_i^f \tilde{A}_i^a {\Gamma_i^f}^T, \\ \tilde{A}_i^a &= \left(\left(Y^f T\right)^T {R_i^l}^{-1} Y^f T + T^T W^{-1} T\right)^{-1}, \\ L_i^a &= L_i \Gamma_i^f, \end{aligned}$$

and

$$\Gamma_i^f = C^f \Omega_r^f {U^f}^T V^{-1} T.$$

This method has same asymptotic computational cost of the SEIK algorithm, but higher order of convergence in the approximations of m_i and h_i.
Furthermore, it takes into account the noise-like effect caused by the model error, that is completely neglected by the SEIK filter.
However, the drawback is that it cannot be blindly applied to any situation. In fact, it necessitates of a case by case analytical study, in order to fasten the computation of y_i^Q and R_i^Q, which is otherwise too expensive.

5.3 Model error estimation

In order to produce a model error (Q_i) estimation, a few remarks are needed. First of all, since Q_i evaluates how much the model system is able to reproduce reality, it cannot be estimated by the model itself. Instead, a comparison with (error-affected) observation can give information about the skill of the model. Thus, a data driven approach is advisable.
However, data information is rarely enough to estimate Q_i if the degrees of freedom are not drastically reduced by some other reasoning, like no correlation between far places, same variance in certain areas, etc.
Furthermore, some kind of temporal connection between the Q_i's is necessary, in order to relate information from the various observation y_i.
The simplest solution is to consider Q_i constant in time, diagonal with just one degree of freedom, i.e.

$$Q_i = Q = q^2 I_N, \quad q \in \mathbb{R}, \tag{5.56}$$

but many other option can be considered, taking into account that more degrees of freedom need more data information.
In biogeochemistry, the form proposed in equation (5.56) is a not bad starting point. In fact, if the system variables are logarithmic concentrations, then this quantifies the uncertainty of the model as a percentage error, that it is more desirable in predator-prey dynamics compared to addictive errors.
That said, the approach presented in this section can be adapted for any chosen Q form, and it is based on a maximum likelihood strategy.
The probability to observe data $y_{0:i}$ can be written

$$p\left(y_{0:i}\right) = p_i^l\left(y_i\right) p\left(y_{0:i-1}\right), \tag{5.57}$$

where

$$p_i^l\left(y_i\right) \coloneqq p\left(y_i | y_{0:i-1}\right)$$

is the likelihood probability at time t_i.
Thus, substituting i by $i-1$ in equation (5.57),

$$p\left(y_{0:i-1}\right) = p_{i-1}^l\left(y_{i-1}\right) p\left(y_{0:i-2}\right), \tag{5.58}$$

and, using equation (5.58) into (5.57),

$$p\left(y_{0:i}\right) = p_i^l\left(y_i\right) p_{i-1}^l\left(y_{i-1}\right) p\left(y_{0:i-2}\right).$$

Proceeding in the same way, by induction over i, the following identity is proved:

$$p\left(y_{0:i}\right) = p_0^l\left(y_0\right) \cdots p_i^l\left(y_i\right). \tag{5.59}$$

Then, for any given $Q \in \mathcal{Q}$, where $\mathcal{Q}$ is the space of all the considered model error covariance matrices, it is possible to quantify the probability to observe a certain data series, and the Q that corresponds to the higher probability is more likely the best approximation of the real model error.
Thus it is possible to write a function f that relates the coordinates of Q (that is to say degrees of freedom) to the logarithm of the probability in equation (5.59). For example, if Q depends on one degree of freedom, like in the above example (equation (5.56)), the function has the form

$$f(q) = -\ln\left(p_0^l(y_0)\right) - \ldots - \ln\left(p_i^l(y_i)\right),$$

where the right hand side likelihood probabilities are calculated using q.
Now, note that f is a positive function, since probabilities cannot be higher than 1, and, if f can be computed numerically in a closed domain, then it is possible to minimize it. If q_0 is the point where the minimum is reached, then the corresponding matrix is the maximum likelihood estimation of the model error covariance.
How to compute the Gaussian approximation

$$p_i^l(y_i) \approx \mathcal{N}\left(y_i; y_i^l, P_i^l\right)$$

is already explained in Section 5.2, where the relevant equations for SEEK and SEIK are (5.27), (5.28), (5.45) and (5.49).
Then, applying the logarithm and omitting multiplicative and addictive constant factors, the logarithm of the likelihood probability can be stored in the new s_i variable at every time step, i.e.

$$s_i := \ln\left|P_i^l\right| + \left(y_i - y_i^l\right)^T P_i^{l^{-1}}\left(y_i - y_i^l\right),$$

where $\left|P_i^l\right|$ is the determinant of P_i^l.
Summarizing, at the cost of filtering one time all the available data, the sum S of the s_i's, i.e.

$$S = \sum_{i=0}^{K} s_i,$$

represents an indicator of the goodness of the Q matrix used. Then, any numerical minimizer (e.g. MATLAB's *fminbnd*) can be used to find the best available covariance matrix Q.

Chapter 6

Experiments

In this chapter, 3D-VAR, SEIK, and improvements to SEIK introduced in Chapter 5 are tested with a twin-experiment.

The modelled system represents the photic zone of a closed square marine system, and it is composed by a biogeochemical seven variables Fasham-like model coupled with a physical advection-diffusion transport model.

The system variables are phytoplankton, zooplankton, bacteria, detritus and three nutrients, while the physical components, like water velocity, are considered as forcing.

The twin experiment consists of a simulated reality, from which error-affected measurements are taken. Then, the Data Assimilation methods are used to estimate reality from observations, and the skill performance of the different methods is evaluated.

The rooted mean square distance indicator is used to compare the skill of the methods.

The results of the experiments indicates that SEIK is greatly superior to 3D-VAR.

The Standard SEIK and the modified (by Sections 5.1 and 5.2) version instead differ in behaviour: while they are near in RMSD values, the latter presented a higher resilience to divergences induced by a large model error.

Finally, a second twin experiment has been used to test the model error estimation procedure described in Section 5.3, successfully obtaining values near to the true ones. The estimations produced by the modified SEIK are more accurate then the estimations produced by the standard filter. In all cases, using the estimated parameter induced a slightly better performance, if compared with the true model error results.

6.1 The biogeochemical model

The biogeochemical model is a seven variables Fasham-like model ([15]), simulating reactions in the photic zone between phytoplankton (P), zooplankton (Z), bacteria (B), detritus (D) and three nutrients, namely nitrate nitrogen (N_n), ammonium nitrogen (N_r) and labile dissolved organic nitrogen or DON (N_d). The biogeochemical model *per se* has no spatial dimensions.
The equations are:

- Phytoplankton concentration P:

$$\frac{dP}{dt} = (1-\gamma)\,J\,(Q_n + Q_r)\,P - G_P - \mu_P P - mP,$$

  where

  - γ is the fraction of total net primary production that is exuded by the phytoplankton as DON.
  - J is the light limited grow rate and it depends on the time t.
  - $Q_n + Q_r$ is the nutrient limiting factor. Q_n depends on both N_n and N_r while Q_r only depends on N_r:

$$Q_n = \frac{N_n e^{-\Psi N_r}}{K_n + N_n},$$

$$Q_r = \frac{N_r}{K_r + N_r},$$

    with K_1 and K_2 half saturation constants for nitrate and ammonium uptakes respectively, and Ψ a constant parametrizing the strength of the ammonium inhibition of nitrate uptake.
  - G_P represents the loss of population due to zooplankton grazing.
  - μ_P is the specific natural mortality factor.
  - m parametrizes the diffusive mixing between the photic zone and the deep layer.

- Zooplankton concentration Z:

$$\frac{dZ}{dt} = \beta_P G_P + \beta_B G_B + \beta_D G_D - \mu_Z Z - \mu_Z^* Z,$$

  where

- G_P, G_B, G_D, represent the zooplankton grazing on phytoplankton, bacteria and detritus. They all have the same form, e.g. G_P can be written as

  $$G_P = gZ\frac{p'_P P}{K_Z + F},$$

  with g maximum specific grazing rate, K_Z is the half saturation constant for grazing, $F = p'_P P + p'_B B + p'_D D$ is the total food and the primed elements are measurements of the zooplankton preferences for the various food types and they depend from the availability, i.e.

  $$p'_P = \frac{p_P P}{p_P P + p_B B + p_D D}$$

  where p_P, p_B and p_D are zooplankton preferences constants.

- β_P, β_B and β_D are the assimilation efficiencies.
- μ_Z is the specific natural mortality factor.
- μ^*_Z is the zooplankton specific excretion rate.

• Bacteria concentration B:

$$\frac{dB}{dt} = U_d + U_r - G_B - \mu^*_B B - mB,$$

where

- U_d and U_r are the DON and ammonium uptake respectively, and can be quantified as

  $$U_d = \frac{V_B B N_d}{K_B + S + N_d}$$

  and

  $$U_r = \frac{V_B B S}{K_B + S + N_d},$$

  with V_b maximum bacterial uptake rate, K_B half saturation coefficient for uptake and

  $$S = \min\{N_r, \eta N_d\}$$

  the total bacteria nitrogenous substrate, where η is the ammonium/DON uptake ratio.

- μ^*_B is the bacterial specific excretion rate.

- Detritus concentration D:

  $$\frac{dD}{dt} = (1-\beta_P)\,G_P + (1-\beta_B)\,G_B - \beta_D G_D - \mu_D D + \mu_P P - mD - VD,$$

  where

  - μ_D is the specific rate of breakdown of detritus to DON.
  - V is the detrital sinking rate.

- Nitrate concentration N_n:

  $$\frac{dN_n}{dt} = -JQ_n P + m\,(N_0 - N_n)\,,$$

  where

  - N_0 is the nitrate concentration below the photic zone. This depends on time and on space, after the coupling with the transport model.

- Ammonium concentration N_r:

  $$\frac{dN_r}{dt} = -JQ_r P - U_r + \mu_B^* B + (\epsilon\mu_Z^* + (1-\Omega)\,\mu_Z)\,Z - mN_r,$$

  where

  - ϵ is the ammonium rate in zooplankton excretion.
  - $(1-\Omega)$ is the remineralization of grazed zooplankton by unmodelled higher predators.

- DON concentration N_d:

  $$\frac{dN_d}{dt} = \gamma J\,(Q_n + Q_r)\,P + \mu_D D + (1-\epsilon)\,\mu_Z^* Z - U_d - mN_d.$$

Accordingly with [15], the parameters have been set as in Table 6.1.
J and N_0 do not appear in the list, since they are not constant. The first one is dependent on time, and follows a seasonal cycle, the last one is dependent on both time and space, following a temporal seasonality and rising with a linear spatial dependency from the left side to the right side of the domain.

Biogeochemical parameters	
$\gamma = 0.05$	$m = 0.001\, d^{-1}$
$\mu_P = 0.045\, d^{-1}$	$\mu_Z = 0.05\, d^{-1}$
$\mu_Z^* = 0.10\, d^{-1}$	$\mu_B^* = 0.05\, d^{-1}$
$\mu_D = 0.05\, d^{-1}$	$\Psi = 1.5\ (N\, mMol)^{-}1$
$K_n = 0.5\, N\, mMol\, m^{-3}$	$K_r = 0.5\, N\, mMol\, m^{-3}$
$K_Z = 1\, N\, mMol\, m^{-3}$	$K_B = 0.5\, N\, mMol\, m^{-3}$
$\beta_P = 0.75$	$\beta_B = 0.75$
$\beta_D = 0.75$	$p_P = 10$
$p_B = 4$	$p_D = 1$
$g = 1.0\, d^{-1}$	$\eta = 0.6$
$V_B = 2.0\, d^{-1}$	$V = 0.1\, m\, d^{-1}$
$\epsilon = 0.75$	$\Omega = 0.33$

Table 6.1: Parameters of the biogeochemical model.

6.2 Coupling with the transport model

The physical model is an advection-diffusion transport model, and it is online coupled with the biogeochemical model. It works on a 2D square domain, and simulates the effects of currents and diffusion on the biogeochemical tracers. The equation is

$$\frac{\partial C}{\partial t} = -v \cdot \nabla C + k\Delta C + \frac{\partial C_b}{\partial t},$$

where C represents the concentration of a tracer, v is the water velocity vectorial field depending on time and space, $k = 500\, m^2\, s^{-1}$ is the diffusivity coefficient and $\dfrac{\partial C_b}{\partial t}$ is the derivative of the tracer obtained from the biogeochemical reactions, solved by the biogeochemical model described in Section 6.1.

The v field is built by a procedure based on random coefficients, that combines sinusoids and second order polynomials in space and time, in order to obtain a smooth field of vectors with module between 0 and $0.5\, m\, s^{-1}$ on average.

All the showed equations has been translated in logarithmic form, as well as the system states. This can be done easily, in fact, if

$$\tilde{x} = \ln x$$

and

$$\frac{dx}{dt} = f\left(x\right),$$

then

$$\frac{d\tilde{x}}{dt} = \frac{d(\ln x)}{dt} = \frac{1}{x}\frac{dx}{dt} = \frac{f(x)}{x} = \frac{f(\exp(\tilde{x}))}{\exp(\tilde{x})}.$$

In this way, concentrations are forced to be positive, avoiding some numerical instability problems. Furthermore, applying a Kalman-Filter method to a logarithmic variable is equivalent to consider log-normal errors on the non-logarithmic version of the variable, which is something desirable when operating with prey-predator dynamics. In fact, such systems usually have cyclical exponential growths and losses that induce errors proportional to the logarithm of the considered quantity.

From the numerical point of view, the coupled model is discretized with a finite volume method, using a mesh of 25 cells (5 × 5), and its integration is managed by the *ode45* MATLAB solver.

An example of one-year dynamic is reported in Figures 6.1 to 6.8

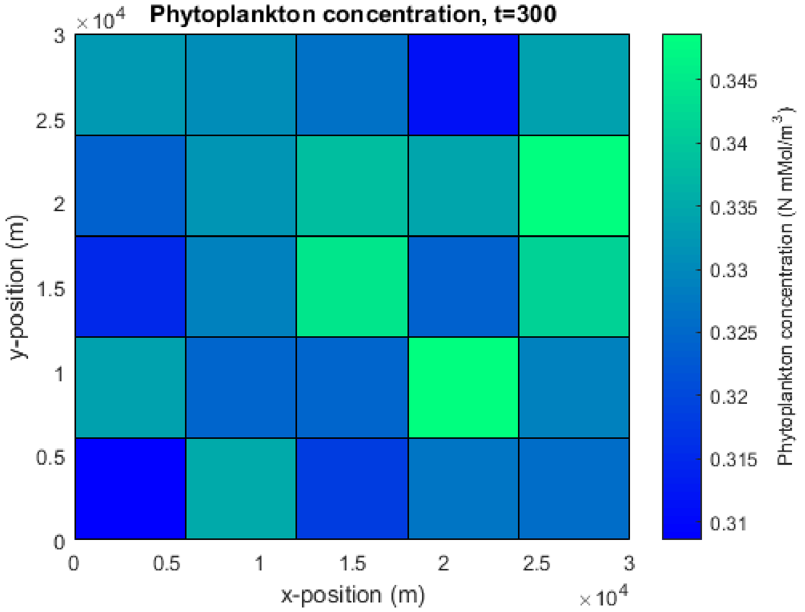

Figure 6.1: Example of phytoplankton concentration at day 300.

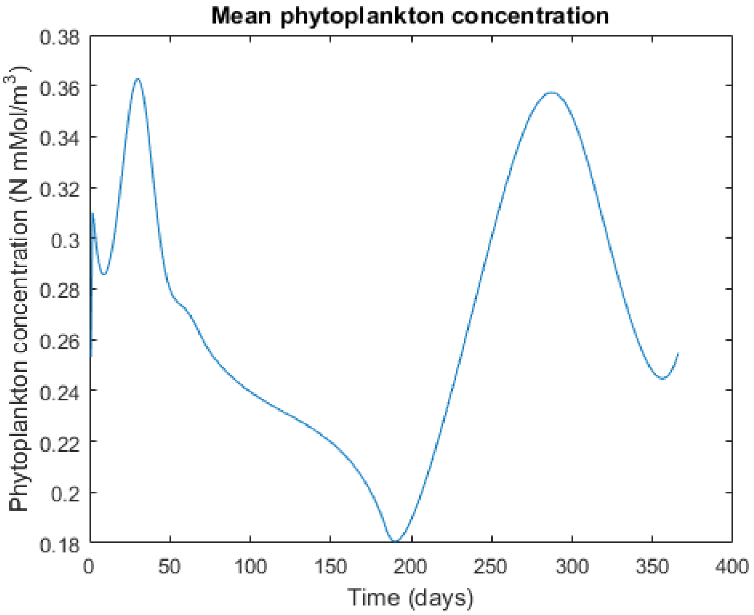

Figure 6.2: Example of phytoplankton concentration: spatial mean by time.

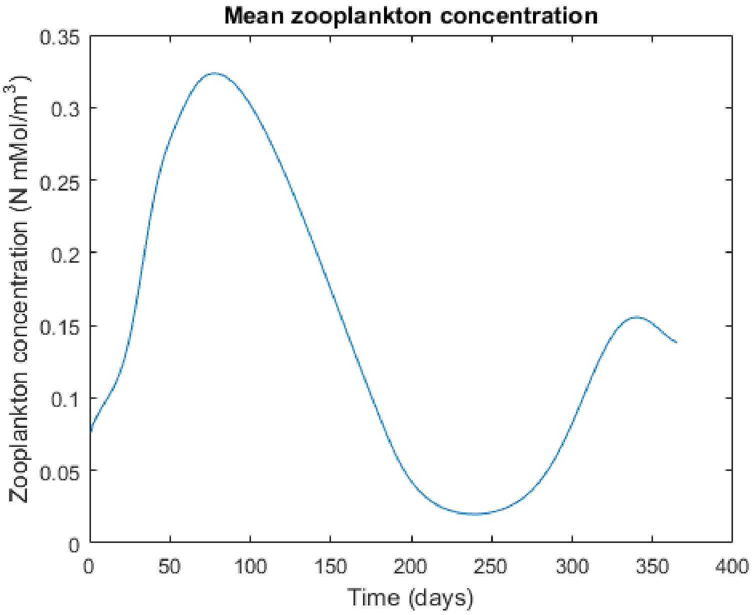

Figure 6.3: Example of zooplankton concentration: spatial mean by time.

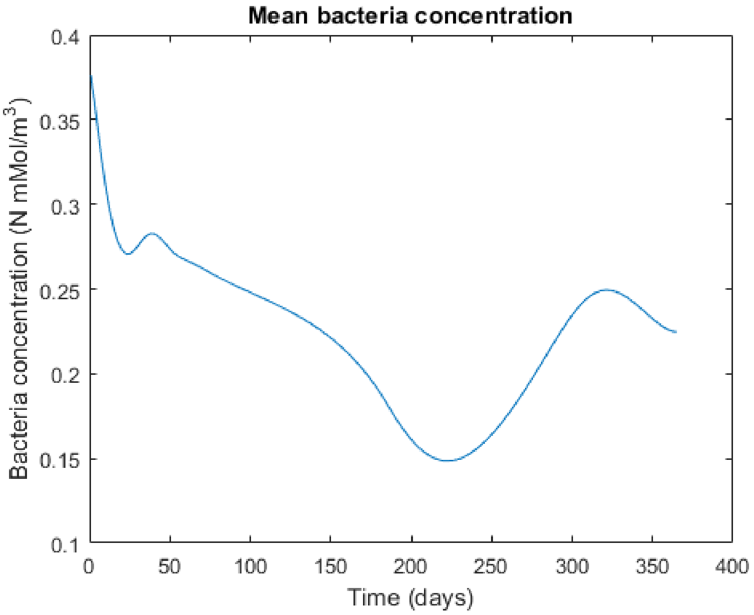

Figure 6.4: Example of bacteria concentration: spatial mean by time.

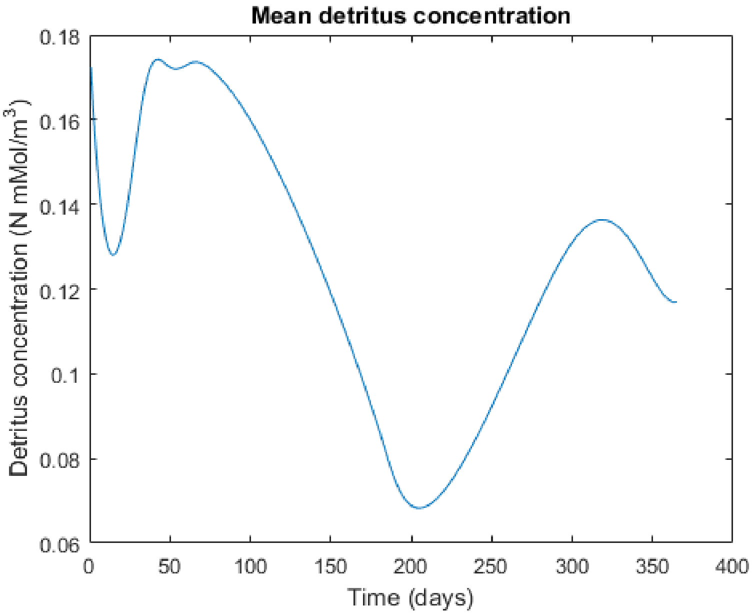

Figure 6.5: Example of detritus concentration: spatial mean by time.

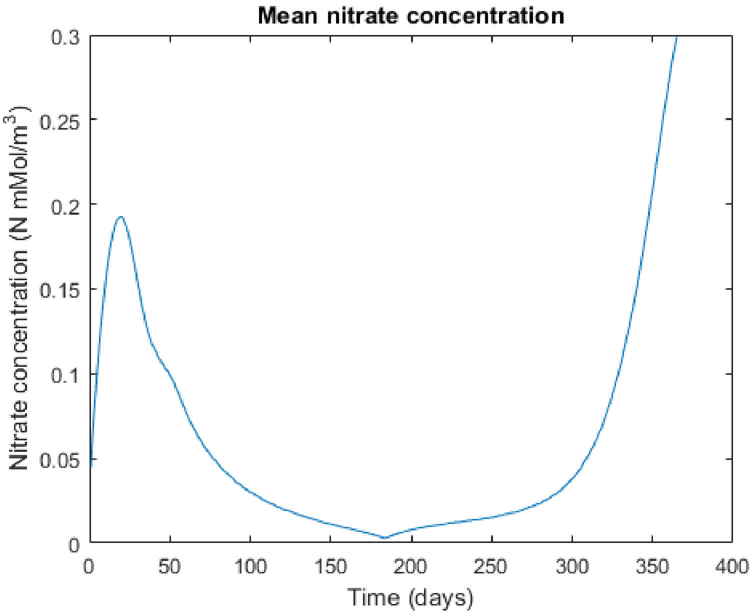

Figure 6.6: Example of nitrate concentration: spatial mean by time.

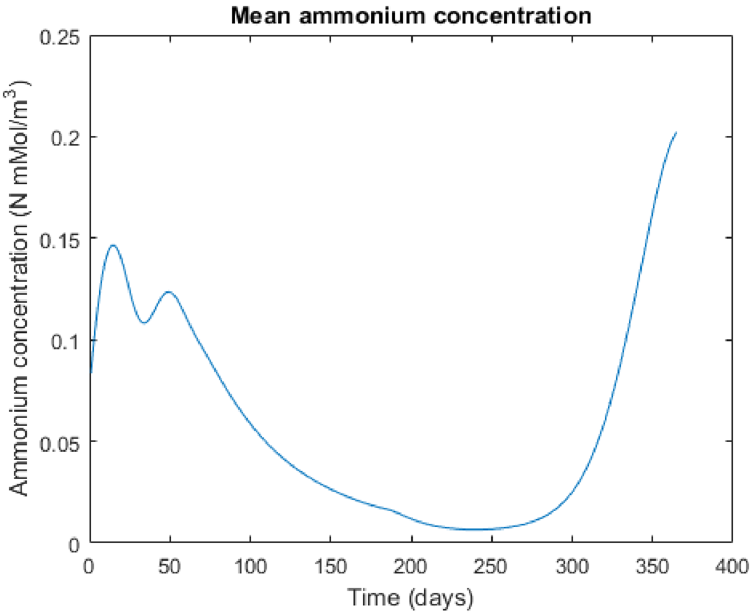

Figure 6.7: Example of ammonium concentration: spatial mean by time.

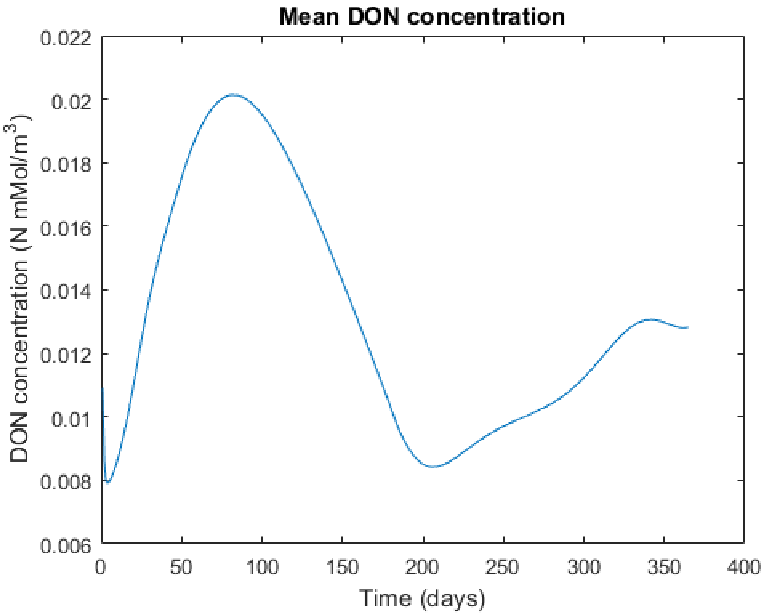

Figure 6.8: Example of DON concentration: spatial mean by time.

6.3 The observation operator

The observation operator simulates a satellite measuring chlorophyll (i.e. phytoplankton) in the whole domain. Two different observation operators have been tested: one with an addictive zero-mean error (with standard deviation of 0.01 $N\, mMol\, m^{-3}$) on the measured concentration, and the other one with a percentage error instead (i.e. standard deviation of 0.1 on the logarithm of the concentration, namely around 7% on the concentration).

6.4 Data Assimilation

Three Data Assimilation schemes have been used, 3D-VAR, standard SEIK and modified SEIK (by strategies proposed in Sections 5.1 and 5.2).
At initialisation purpose, monthly means and PCA most relevant covariance components have been used as starting point for all the algorithms (to produce the initial state error covariance decomposition in SEIK's, and to build the B matrix replacing Q_i, as prescribed in [44]). PCA and means computation has been done based on a total of 200 years of simulation, coming from a 10-years run for each one of 20 different random forcings (i.e. the velocity fields described in Section 6.2).

In both the SEIK-based systems, 3 different reduced error space dimensions has been tested, namely $r = 2$, 6 and 12, corresponding to an ensemble size of 3, 7 and 13 members. These choices come from the necessity to solve a non linear system in order to use a high order exact sampling (see Section 5.1), and, at those dimensions, the solution is easily found.
In order to apply the modifications in Section 5.2, a cheap computation of y^Q and R^Q is needed. This is possible thanks to the simple form of the observation operator and by choosing a diagonal model error Q.
In fact, since both the observation operators are of the form

$$h : \mathbb{R}^N \longrightarrow \mathbb{R}^n$$

$$h \begin{pmatrix} x_1 \\ \vdots \\ x_N \end{pmatrix} = \begin{pmatrix} h_1\left(x_{j_1}\right) \\ \vdots \\ h_n\left(x_{j_n}\right) \end{pmatrix},$$

with $h_i : \mathbb{R} \longrightarrow \mathbb{R}$ for every $i \in \{1, \dots, n\}$ and $\{j_1, \dots, j_n\} \subseteq \{1, \dots, N\}$, then, using the ensemble produced by the third order exact U and v

$$U = \left(\begin{array}{c} \frac{1}{\sqrt{2}} I_N \\ \hline -\frac{1}{\sqrt{2}} I_N \end{array} \right),$$

$$v = \frac{1}{\sqrt{2N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix},$$

any ensemble member differs from any other by at most one coordinate. Thus, the action of h to the whole ensemble can be computed by only calculating exactly $3n$ scalar unidimensional functions h_i.

6.5 The twin experiment

The twin experiment works in the following way:

- The "reality" is prepared as follows:

  - first, starting from an arbitrary initial state (picked randomly around January monthly mean), the model operator is used to evolve the system state to the following day. Then, a zero mean Gaussian error with covariance Q is added in order to represent model errors. This sum is referred to as the "real state" of the system at day 1. Note that the real state ideally includes sub-grid processes and other not modelled events.
  - In order to preserve a realistic evolution, the initial condition for next day computation is obtained by projecting the real state into the nearest state that would fit well with the simulated model dynamics. This means that the portion of the summed error not proportional to the monthly variability of the system is neglected. Note that the variability of the system has been evaluated in terms of principal components of the PCA obtained by the 200-years simulation used in Section 6.4.
  - The model operator is used to evolve the system state to the following day, a zero mean Gaussian error with covariance Q is added and the whole procedure is repeated until one year of simulation is computed.

  Various different Qs have been tested, in the form

$$Q = q^2 I_N,$$

  as in equation (5.56). The values chosen for q are 0.05, 0.15 and 0.25, representing around 3%, 10% and 20% percentage error, and later referred to as "small", "medium" and "large" model error respectively.

- Observations are computed in the following way:
  - every 7 days the observation operator is applied to the real state.
  - The result is then added to a zero mean Gaussian error with covariance R, as explained in Section 6.3, in order to account for the measurement errors.
- Each Data Assimilation scheme is initialized at time t_0 in the same state, as explained in Section 6.4.
- Each Data Assimilation scheme computes its forecast and analysis for the whole year.
- The results are compared in order to study the skill of each method.

6.6 The RMSD metric

Summarizing, a total of 210 one-year experiments have been ran, varying several elements: 5 different random initial conditions and forcings (as described in Sections 6.5 and 6.2 respectively), 2 observation operators (see Section 6.3) and 3 model errors (represented by a diagonal covariance matrix as presented in Sections 6.5) for each one of the 3 Data Assimilation schemes. Furthermore, in the SEIK-based experiments, 3 different ensemble sizes as been tested, as explained in Section 6.4.
In order to compare them all, the simulations have been grouped by model error for each Data Assimilation scheme, and the root mean square deviation (RMSD) between reality and analysis has been chosen as performance metric. The RMSD has been applied to the logarithm of all the concentration variables of all the experiment in each group. In this way, the RMSD is used as indicator of the relative error affecting the scheme, and does not need any normalization. In fact, after the application of the logarithm, differences represent concentration ratios, and the result does not depend by unities of measure or magnitudes.

6.7 Results

The obtained results are presented in this section in aggregated form, mainly averaged by the RMSD indicator, as explained in Section 6.6. Each comparison is discussed separately.

6.7.1 3D-VAR vs SEIK

In all the tests the SEIK filter has a better performance.
Differently from SEIK, all the 3D-VAR runs not set with the lowest model error produced a divergence, usually during the last part of the year, where the dynamics are faster. In the low model error settings, the comparison in terms of RMSD of the two methods shows that the variational method has 10 times the RMSD of the Kalman-Filter.

	3D-VAR	**SEIK**
Small model error	0.899	0.080
Medium model error	diverged	0.200
Large model error	diverged	0.324

Ensemble size = 3	**SEIK**	**Modified SEIK**
Small model error	0.082	0.081
Chl-a	0.064	0.064
Others	0.085	0.084
Medium model error	0.213	0.202
Chl-a	0.160	0.160
Others	0.221	0.208
Large model error	0.350	0.321
Chl-a	0.262	0.263
Others	0.363	0.330

Table 6.3: SEIK – modified SEIK with 3 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported.

Table 6.2: 3D-VAR – SEIK: RMSD comparison by model error magnitude.

6.7.2 SEIK vs modified SEIK

The two schemes are quite similar in term of performances, mainly in case of small model error or large ensemble size (less then 1% difference of RMSD). However, in the smaller ensemble size case (Table 6.3), the medium and large model error settings lead to a performance gain (up to 10% better RMSD) of the modified SEIK over the classic one.

This behaviour is compatible with the theoretical derivation of the improvements presented in Sections 5.1 and 5.2.

In fact, the better convergence offered by the higher order exact sampling can help a smaller ensemble with a better approximation of means and covariances.

Moreover, a large model error introduce stronger fluctuations that cannot be managed by the classical SEIK algorithm, which is instead the main purpose of the modification presented in Section 5.2.

A possible explanation about the fact that these differences does not show up in bigger ensemble size cases (like in Table 6.5), is that, with a reduced error space dimension larger then 6, the majority of the variability has already been taken into account. This reasoning is supported by the graphs in Figure 6.9 , that represent the standard deviations of the first 20 PCA components

Ensemble size = 7	**SEIK**	**Modified SEIK**
Small model error	0.079	0.079
Chl-a	0.064	0.064
Others	0.097	0.096
Medium model error	0.196	0.197
Chl-a	0.160	0.160
Others	0.201	0.202
Large model error	0.323	0.321
Chl-a	0.261	0.261
Others	0.332	0.330

Table 6.4: SEIK – modified SEIK with 7 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported.

Ensemble size = 13	**SEIK**	**Modified SEIK**
Small model error	0.080	0.080
Chl-a	0.063	0.063
Others	0.082	0.082
Medium model error	0.204	0.202
Chl-a	0.159	0.159
Others	0.211	0.208
Large model error	0.329	0.330
Chl-a	0.260	0.260
Others	0.339	0.340
Special case	0.497	0.421
Chl-a	0.287	0.273
Others	0.524	0.441

Table 6.5: SEIK – modified SEIK with 13 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported. The last 3 lines represent a particular simulation made at large model error, referred as “special case”.

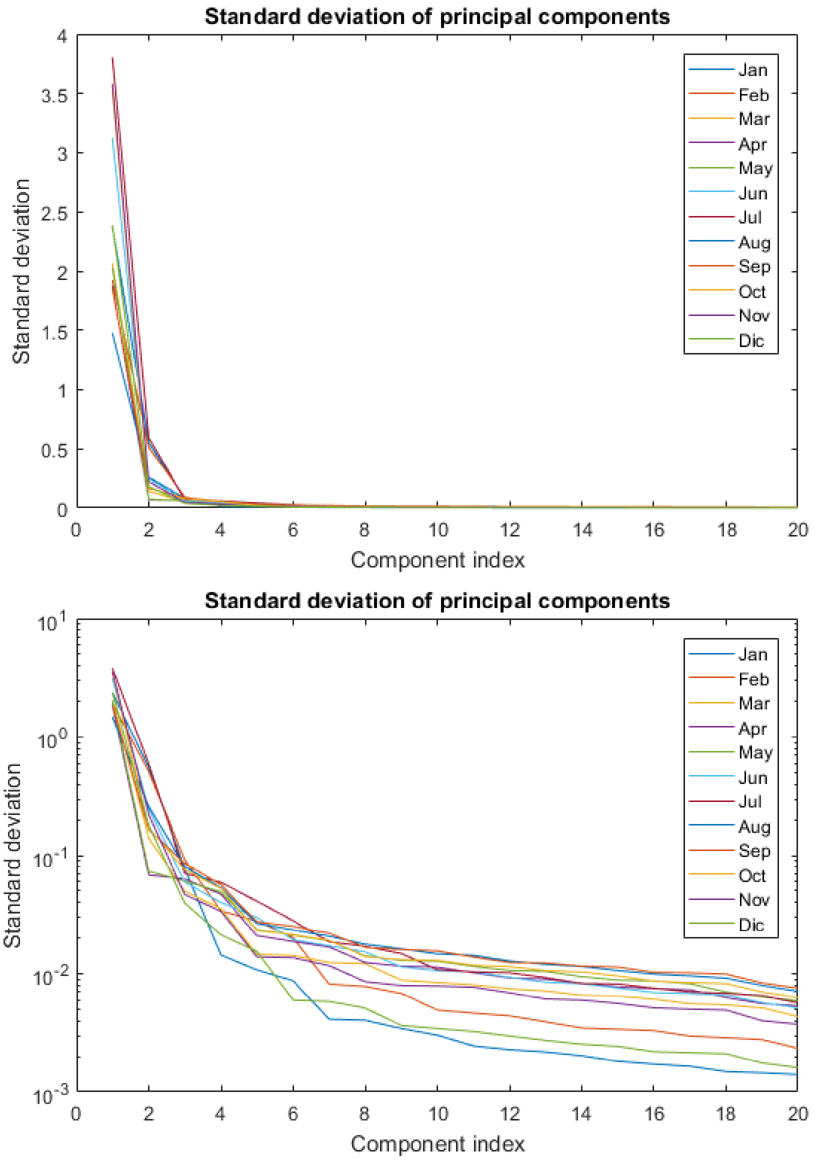

Figure 6.9: Standard deviations associated to the first 20 principal components of the monthly variability of the system, in decimal and logarithmic scale.

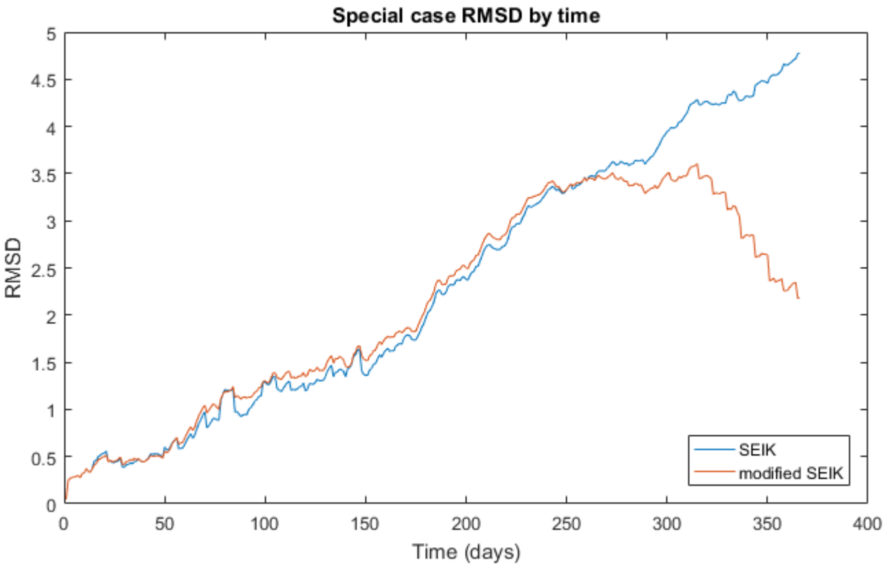

Figure 6.10: Special case RMSD by time. Blue is for classic SEIK, red for modified SEIK.

of the system, month by month, in decimal and logarithmic scale. From the pictures is clear that the first 6 components are enough for picturing the variability of the system up to the 2nd significant digit, i.e. the 99%.

In such a situation, where the vast majority of the fluctuations are already taken into account, the modifications related to the model error are less or not relevant at all. In the same way, if the number of ensemble members is high enough, the higher convergence order of the sampling method of the modified SEIK is not needed in order to reach the desired precision.

Thus, the modifications to the SEIK algorithm improve the performance of the filter mainly when the situation is more “difficult”.

Another evidence comes from a particular simulation, made at large model error, with 13 ensemble members (conditions where usually the two tested SEIK filters have very similar behaviour), that is referred to as the “special case” in the following. Even if a single example is not statistically significant by itself, it deserves to be mentioned in this context. As shown in Figure 6.10, in the special case both filters start loosing the true trajectory, but the modified version is able to recover faster, scoring a 15% better RMSD (see Table 6.5). In this example, the deviation from the reality induced by the large model error is difficultly managed by the SEIK scheme, while the modified version can better take into account the error effects, as explained in Section 5.2.

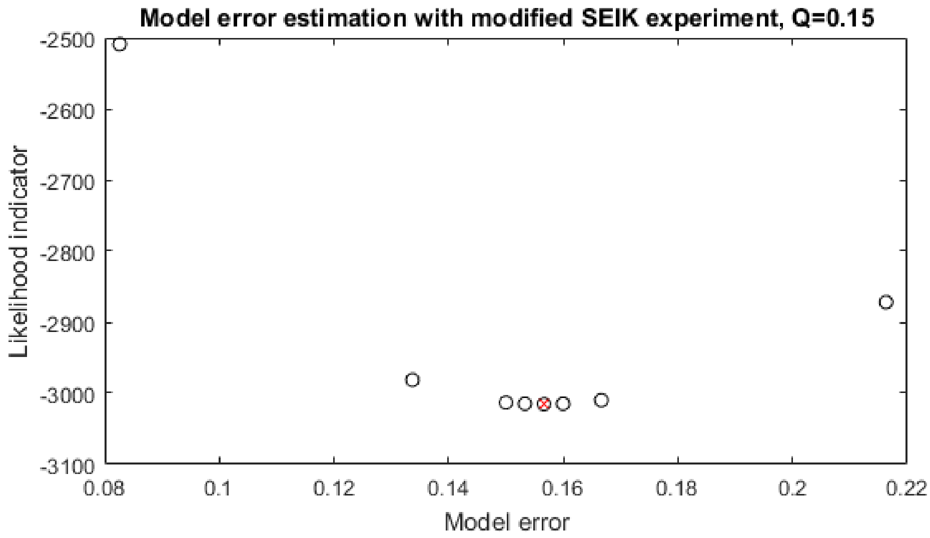

Figure 6.11: Model error estimation with modified SEIK, true model error is 0.15. One black circle is represented for each iteration, the red cross identifies the chosen minimum.

Note that Tables 6.3, 6.4 and 6.5 show, for each group, the total RMSD, as well as its split values from the observed variable (i.e. the chlorophyll) and from all the other variables. As expected, it is easier to estimate the variable for which we have observations, and both the filters obtain very similar results. This means that, in the cases where the modified SEIK achieves a better estimation, the majority of the improvements come from the non-observed variables.

6.8 Model error estimation

A second twin experiment has been prepared in order to test the maximum likelihood technique exposed in Section 5.3.

In this case, a total of 10 experiments has been launched, changing 5 different values of q (namely 0.05, 0.10, 0.15, 0.20, 0.25) and using both the standard SEIK and the modified version.

The likelihood indicator has been minimized using the MATLAB function *fminbnd*, and the solution is found in less than 7 iterations on average. One iteration implies one-year filtering using all the available data.

An example of one experiment is represented in Figure 6.11.

As shown in Tables 6.6 and 6.7, both methods obtained results comparable

SEIK experiments				
Model error q	**Estimation** $\tilde{q}$	**Iterations**	**SEIK at** q	**SEIK at** $\tilde{q}$
0.05	0.054	9	0.097	0.096
0.10	0.139	7	0.269	0.257
0.15	0.219	7	0.549	0.519
0.20	0.267	3	1.007	0.950
0.25	0.293	8	1.341	1.329

Table 6.6: Model error estimation experiments using the standard SEIK. Model error q: the true model error. Estimated $\tilde{q}$: the estimated model error. Iterations: number of iteration needed to obtain the estimation; one iteration involves the assimilation of the entire dataset. SEIK at q: RMSD skill (low is better) of the SEIK method set with the true parameter q. SEIK at $\tilde{q}$: RMSD skill (low is better) of the SEIK method set with the estimated parameter $\tilde{q}$.

Modified SEIK experiments				
Model error q	**Estimation** $\tilde{q}$	**Iterations**	**Modified SEIK at** q	**Modified SEIK at** $\tilde{q}$
0.05	0.054	6	0.096	0.095
0.10	0.103	4	0.269	0.268
0.15	0.156	6	0.551	0.550
0.20	0.212	5	1.007	0.994
0.25	0.285	9	1.382	1.347

Table 6.7: Model error estimation experiments using the modified SEIK. Model error q: the true model error. Estimation $\tilde{q}$: the estimated model error. Iterations: number of iteration needed to obtain the estimation; one iteration involves the assimilation of the entire dataset. Modified SEIK at q: RMSD skill (low is better) of the modified SEIK method set with the true parameter q. Modified SEIK at $\tilde{q}$: RMSD skill (low is better) of the modified SEIK method set with the estimated parameter $\tilde{q}$.

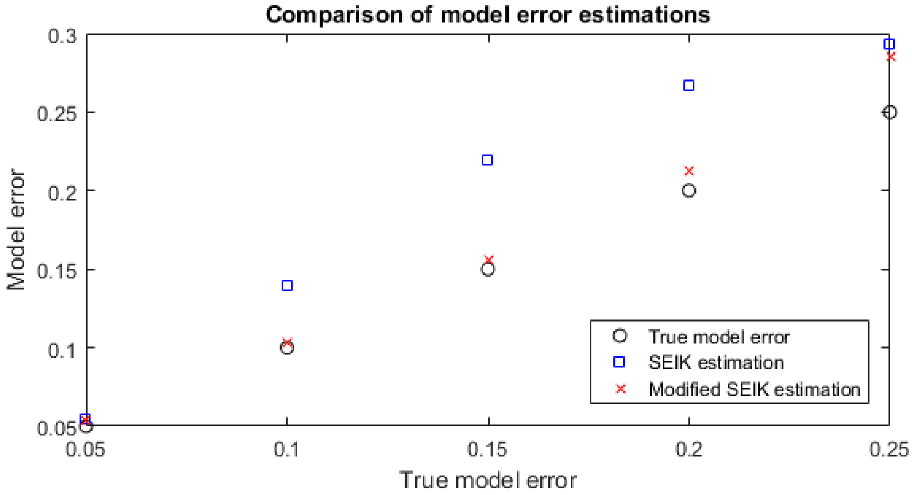

Figure 6.12: Comparison of SEIK and modified SEIK model error estimations.

SEIK vs Modified SEIK model error estimation				
Model error q	**SEIK Estimation**	**Estimation error (%)**	**Modif. SEIK Estimation**	**Estimation error(%)**
0.05	0.054	8%	0.054	8%
0.10	0.139	39%	0.103	3%
0.15	0.219	46%	0.156	4%
0.20	0.267	33%	0.212	6%
0.25	0.293	17%	0.285	14%

Table 6.8: SEIK vs Modified SEIK model error estimation. Model error: the true q model error value. SEIK Estimation: the model error estimated using the SEIK filter. Estimation error (%): the percentage error of the estimation with respect to the true value. Modified SEIK estimation: the model error estimated using the modified SEIK filter.

with the true value of q, but Chapter 5 modifications lead to much more precise results (i.e. values of the estimation $\tilde{q}$ much closer to the model error q). In fact, this procedure applied to the modified SEIK takes advantage of the higher order of convergence of the assimilation scheme and, by fully taking into account the model error, it achieves a better estimation of the likelihood probability. As a consequence, the estimation of the model error obtained with the modified SEIK is always better compared to standard SEIK and up to one order of magnitude better in 2 out of 5 cases (see in particular the comparison between the "Estimation error (%)" columns in Table 6.8 and Figure 6.12).

It is interesting to observe that, in all tests, the RMSD skill of the Data Assimilation is slightly better when launched with the estimated model error values than using the true exact value.

Chapter 7

The smoother

The theoretical considerations presented in Chapter 5 that led to the modified SEIK scheme, can be further expanded in order to complete the novel filter with its own smoother.

Instead of focusing on the estimation of the present state only (namely the analysis state), a smoother is a filtering-like procedure going backward in time, in order to correct past analysis states, propagating the information brought by data not available at the time of the analysis. The new estimation of a past state is called reanalysis.

In this chapter, the smoother is derived theoretically and tested with a twin-experiment.

While, in literature, it is less common then filtering, taking advantage from the smoothing procedure is often advisable, because, as in this case, it can significantly improve the skill of the data assimilation in a certain time window, with a very limited computational effort.

7.1 Derivation

The derivation will follow the same notations used in the previous chapters. In particular, as in Chapter 3, the reanalysis probability p_i^r at the time t_i is defined as

$$p_i^r\left(x_i\right) \coloneqq p\left(x_i|y_{0:K}\right) \tag{7.1}$$

and, by the Law of total probability (Theorem 3),

$$p_i^r\left(x_i\right) = \int_{\mathbb{R}^N} p\left(x_i|x_{i+1}, y_{0:K}\right) p\left(x_{i+1}|y_{0:K}\right) dx_{i+1}. \tag{7.2}$$

By Bayes Theorem (Corollary 2),

$$p\left(x_i|x_{i+1}, y_{0:K}\right) \propto p\left(y_{(i+1):K}|x_i, x_{i+1}, y_{0:i}\right) p\left(x_i|x_{i+1}, y_{0:i}\right)$$

and then

$$p\left(x_i|x_{i+1},y_{0:K}\right)=p\left(x_i|x_{i+1},y_{0:i}\right) \tag{7.3}$$

because $p\left(y_{(i+1):K}|x_i,x_{i+1},y_{0:i}\right)$ does not depend from x_i.
In fact, by Law of total probability,

$$\begin{aligned}
p\left(y_{(i+1):K}|x_{i:(i+1)},y_{0:i}\right) &= \\
&= \int_{\left(\mathbb{R}^N\right)^{K-i-1}} p\left(y_{(i+1):K}|x_{i:K},y_{0:i}\right) p\left(x_{(i+2):K}|x_{i:(i+1)},y_{0:i}\right) dx_{(i+2):K} \\
&= \int_{\left(\mathbb{R}^N\right)^{K-i-1}} p\left(y_{i+1}|x_{i:K},y_{0:i}\right)\cdots p\left(y_K|x_{i:K},y_{0:(K-1)}\right)\cdot \\
&\qquad \cdot p\left(x_{i+2}|x_{i:(i+1)},y_{0:i}\right)\cdots p\left(x_K|x_{i:(K-1)},y_{0:i}\right) dx_{(i+2):K} \\
&= \int_{\left(\mathbb{R}^N\right)^{K-i-1}} p_{\mathcal{H}_{i+1}(x_{i+1})}\left(y_{i+1}\right)\cdots p_{\mathcal{H}_K(x_K)}\left(y_K\right)\cdot \\
&\qquad \cdot p_{\mathcal{M}_{i+2}(x_{i+1})}\left(x_{i+2}\right)\cdots p_{\mathcal{M}_K(x_{K-1})}\left(x_K\right) dx_{(i+2):K},
\end{aligned}$$

where the last equality comes from hypothesis (3.1) and (3.2).
Using Bayes Theorem in equation (7.3), and by hypothesis (3.1) and (3.2),

$$p\left(x_i|x_{i+1},y_{0:K}\right)=\frac{p\left(x_{i+1}|x_i,y_{0:n}\right)p\left(x_i|y_{0:n}\right)}{p\left(x_{i+1}|y_{0:n}\right)}=\frac{p_{\mathcal{M}_{i+1}(x_i)}\left(x_{i+1}\right)p_i^a\left(x_i\right)}{p_{i+1}^f\left(x_{i+1}\right)}. \tag{7.4}$$

Substituting p_i^a with the generic notation p_i^b and by Kalman-Filter's equation (3.36), equation (7.4) can be approximated

$$p\left(x_i|x_{i+1},y_{0:K}\right)=p_i^p\left(x_i\right), \tag{7.5}$$

where, writing equations (3.35) and (3.37) using Section 5.2 formalism,

$$p_i^p\left(x_i\right):=\mathcal{N}\left(x_i;x_i^p,P_i^p\right), \tag{7.6}$$

$$P_i^p:=L_i\Gamma_i^bA_i^p\left(L_i\Gamma_i^b\right)^T, \tag{7.7}$$

$$A_i^p:=\left({L_{i+1}}^TQ_{i+1}^{-1}L_{i+1}+\left(T^TW^{-1}T\right)^{-1}\right)^{-1}, \tag{7.8}$$

$$x_i^p:=x_i^b+L_i\Gamma_i^bA_i^p{L_{i+1}}^TQ_{i+1}^{-1}\left(x_{i+1}-x_{i+1}^f\right) \tag{7.9}$$

and Γ_i^b is the change of variable in equation (5.53), transforming L_i into X^bT, i.e.

$$\Gamma_i^b=C^b\Omega_r^b{U^b}^TV^{-1}T.$$

By definition (7.1) and equations (7.2) and (7.5),

$$p_i^r(x_i) = \int_{\mathbb{R}^N} p_i^p(x_i)\, p_{i+1}^r(x_{i+1})\, dx_{i+1}. \tag{7.10}$$

Equation (7.10) defines a recursive relation, that can be computed sequentially from the last time step to the first.
Noting that analysis and reanalysis probabilities must be the same in the last time step (see definitions (3.4) and (7.1)), then, by equation (5.55),

$$p_K^r = \mathcal{N}\left(x_K; x_K^r, P_K^r\right),$$

where

$$x_K^r = x_K^a,$$
$$P_K^r = L_K A_K^r {L_K}^T$$

and

$$A_K^r = A_K^a.$$

Now, by induction over i, p_i^r is a Gaussian such that

$$p_i^r = \mathcal{N}\left(x_i; x_i^r, P_i^r\right), \tag{7.11}$$

where

$$x_i^r = x_i^b + L_i \Gamma_i^b A_i^p {L_{i+1}}^T Q_{i+1}^{-1}\left(x_{i+1}^r - x_{i+1}^f\right),$$
$$P_i^r = L_i A_i^r {L_i}^T$$

and

$$A_i^r = \Gamma_i^b\left(A_i^p + A_i^p {L_{i+1}}^T Q_{i+1}^{-1} L_{i+1} A_{i+1}^r {L_{i+1}}^T Q_{i+1}^{-1} L_{i+1} A_i^p\right) \Gamma_i^{bT}. \tag{7.12}$$

In fact, using the changes of variables

$$x_i = \left(x_i^b + L_i \Gamma_i^b A_i^p {L_{i+1}}^T Q_{i+1}^{-1}\left(x_{i+1}^r - x_{i+1}^f\right)\right) + L_i z_i$$

and

$$x_{i+1} = x_{i+1}^r + L_{i+1} z_{i+1},$$

the integrated term in equation (7.10) becomes, by equations (7.6), (7.7), (7.8), and (7.9),

$$\begin{aligned} &p_i^p(x_i)\, p_{i+1}^r(x_{i+1}) = \\ &\quad = \mathcal{N}\left(z_i; \Gamma_i^b A_i^p {L_{i+1}}^T Q_{i+1}^{-1} L_{i+1} z_{i+1}, \Gamma_i^b A_i^p \Gamma_i^{bT}\right) \mathcal{N}\left(z_{i+1}; 0, A_{i+1}^r\right). \end{aligned} \tag{7.13}$$

Ensemble size = 3	**Modified SEIK**	**Smoother**
Small model error	0.081	0.056
Chl-a	0.064	0.051
Others	0.084	0.057
Medium model error	0.202	0.169
Chl-a	0.160	0.152
Others	0.208	0.172
Large model error	0.321	0.292
Chl-a	0.263	0.253
Others	0.330	0.298

Table 7.1: Modified SEIK – Smoother with 3 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported.

Using Theorem 5 in equation (7.13) and integrating equation (7.10), it holds that

$$p_i^r(x_i) = \mathcal{N}(z_i; 0, A_i^r),$$

where A_i^r is defined in equation (7.12).
Finally, equation (7.11) is obtained changing variable back, and the induction is proved.
It is interesting to note that this procedure is computationally very cheap, it involves only products of small dimension matrices, and do not need any evaluation of the model operator m_i.

7.2 Tests

For testing purpose, the same twin experiment presented in Chapter 6 has been used in order to assess the skill of the smoother, compared to the filter. The smoothing procedure sensibly improves the overall skill of the method, as can be seen in Tables 7.1, 7.2 and 7.3, leading to a global RMSD improvement between 10% and 30%. All variables take advantage of the smoothing, observed and not observed ones.
As noted in Section 7.1, analysis and reanalysis have different values only in past times, while they are the same in the present time, thus the added precision is not useful to forecasting purpose. Figure 7.1 gives an example of the smoother effect by time.

Ensemble size = 7	**Modified SEIK**	**Smoother**
Small model error	0.079	0.055
Chl-a	0.064	0.050
Others	0.096	0.056
Medium model error	0.197	0.170
Chl-a	0.160	0.150
Others	0.202	0.173
Large model error	0.321	0.301
Chl-a	0.261	0.251
Others	0.330	0.308

Table 7.2: Modified SEIK – Smoother with 7 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported.

Ensemble size = 13	**Modified SEIK**	**Smoother**
Small model error	0.080	0.055
Chl-a	0.063	0.050
Others	0.082	0.56
Medium model error	0.202	0.169
Chl-a	0.159	0.150
Others	0.208	0.172
Large model error	0.330	0.291
Chl-a	0.260	0.250
Others	0.340	0.297

Table 7.3: Modified SEIK – Smoother with 13 ensemble members: RMSD comparison by model error magnitude. In each group, the RMSD of the observed variable and other variables is reported.

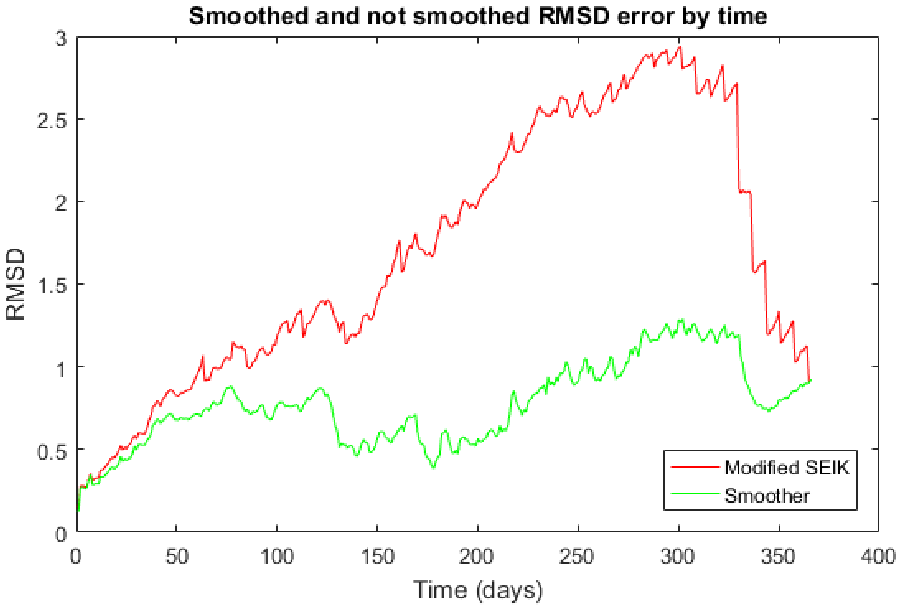

Figure 7.1: Smoother effect example. Red is for modified SEIK, green is for the smoother

Chapter 8

3D realistic implementation

This chapter shows an implementation of the modified SEIK algorithm in a realistic 3D marine model, namely the OGSTM-BFM, which is responsible of forecasting the biogeochemical state of the Mediterranean Sea in the Copernicus Marine Environment Monitoring Service (CMEMS).
The main goal is to demonstrate the feasibility of such an implementation, showing an example of assimilation of real satellite observations of surface chlorophyll.
Furthermore, Section 8.4 presents the parallelization strategies used to rise efficiency.

8.1 The OGSTM-BFM model

The OGSTM-BFM system is a physical 3D transport model coupled with a biogeochemical 1D model.
In particular, OGSTM simulates tracers transport in the Mediterranean Sea due to advection-diffusion processes, while BFM computes the reactions in the water column, taking into account 51 biogeochemical variables (including various kinds of phytoplankton, zooplankton, bacterioplankton, nutrients and organic matter), as overviewed by Figure 8.1.
Due to the complexity of the system, a detailed description is not included in this work, but an interested reader can find it in literature (for example [1], [45], [28] etc).

8.2 Data Assimilation

The assimilation experiment covers a 1 year period, from 1 January to 31 December 2013.

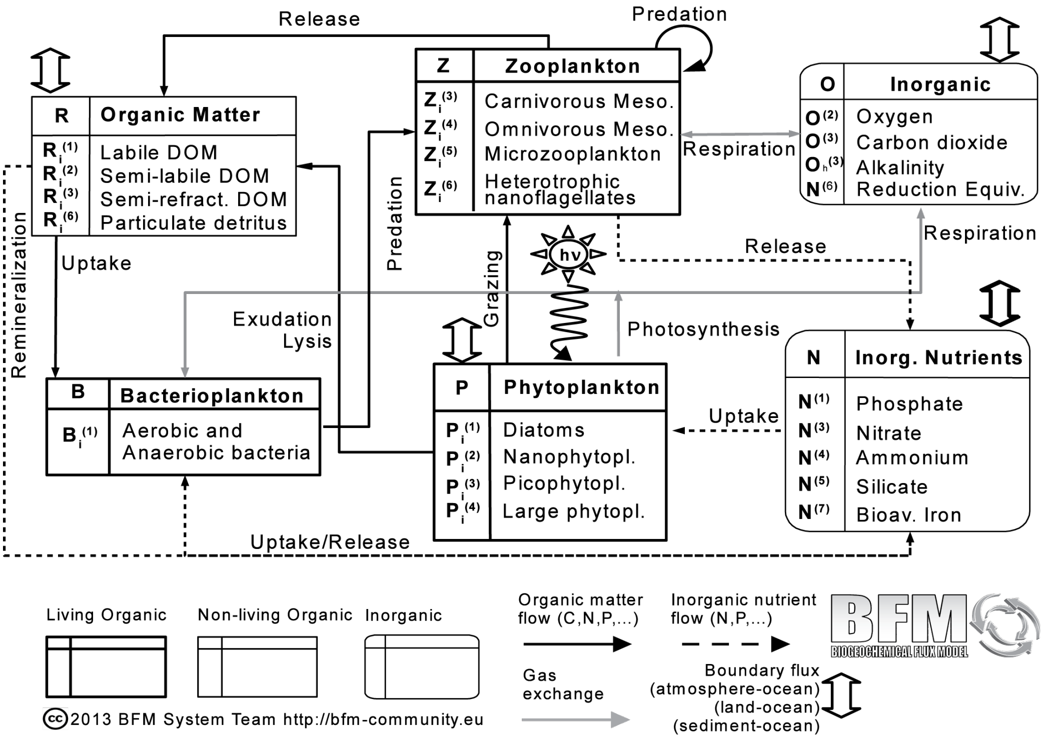

Figure 8.1: BFM variables

The surface chlorophyll concentration used in the assimilation scheme is a product of the CMEMS Ocean Colour Thematic Assembly Centre (OC-TAC) service obtained using remote sensing reflectance (Rrs) spectra.
The system has been tuned accordingly with [45], except that the resolution used is $\frac{1}{4}^o$. The model error $Q = q^2 I_N$ on the logarithm of concentrations has been set at $q = 0.3$, corresponding to around 35% percentage error, and the ensemble size has been fixed at 13 members.
The observation operator h measures the superficial chlorophyll summing the concentrations of the 4 chlorophyll variables (P1l, P2l, P3l, P4l) in the top cell of the water column.
Thus, if $x_{i,j,k,c}$ is a coordinate of the system state representing the concentration of tracer c in the cell with coordinates i, j, k, then

$$h : \mathbb{R}^N \longrightarrow \mathbb{R}^n$$

$$h \begin{pmatrix} \vdots \\ x_{i,j,k,c} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l}, x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l}\right) \\ \vdots \end{pmatrix},$$

where

$$\tilde{h} : \mathbb{R}^4 \longrightarrow \mathbb{R}$$

$$\tilde{h}\left(c_1, c_2, c_3, c_4\right) = \ln\left(c_1 + c_2 + c_3 + c_4\right).$$

In order to use the improvements in Section 5.2, an estimation of y^Q and R^Q (as defined in that section, with the time index here omitted) must be provided. This can be accomplished by noting that a third order exact sampling procedure can be reduced to a few simple calculations if U and v have the form

$$U = \left(\begin{array}{c} \frac{1}{\sqrt{2}} I_N \\ \hline -\frac{1}{\sqrt{2}} I_N \end{array} \right)$$

and

$$v = \frac{1}{\sqrt{2N}} \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

In fact, any exact sampling member member differs from any other by at most one coordinate. Thus, the action of h on the whole ensemble can be computed by exactly only $9n$ evaluations of the the scalar function $\tilde{h}$.
After all simplifications,

$$y^Q = \frac{A_1 + \ldots + A_4 + a_1 + \ldots + a_4 + (2N-8)\, b}{2N},$$

where $A_1, \ldots, A_4, a_1, \ldots, a_4, b \in \mathbb{R}^n$ such that

$$A_1 = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l} \exp\left(\sqrt{N}q\right), x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l}\right) \\ \vdots \end{pmatrix},$$

$$\vdots$$

$$A_4 = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l}, x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l} \exp\left(\sqrt{N}q\right)\right) \\ \vdots \end{pmatrix},$$

$$a_1 = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l} \exp\left(-\sqrt{N}q\right), x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l}\right) \\ \vdots \end{pmatrix},$$

$$\vdots$$

$$a_4 = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l}, x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l} \exp\left(-\sqrt{N}q\right)\right) \\ \vdots \end{pmatrix},$$

$$b = \begin{pmatrix} \vdots \\ \tilde{h}\left(x_{i,j,1,P1l}, x_{i,j,1,P2l}, x_{i,j,1,P3l}, x_{i,j,1,P4l}\right) \\ \vdots \end{pmatrix}.$$

In order to compute R^Q, the residuals $\tilde{A}_1, \ldots, \tilde{A}_4, \tilde{a}_1, \ldots, \tilde{a}_4, \tilde{b} \in \mathbb{R}^n$ are obtained by subtracting y^Q to the above corresponding vectors. Then, the variance, i.e. the diagonal of the R^Q matrix, can be written as

$$\mathrm{diag}\left(\mathrm{R^Q}\right) = \frac{\mathrm{\tilde{A}_1^2 + \ldots + \tilde{A}_4^2 + \tilde{a}_1^2 + \ldots + \tilde{a}_4^2 + (2n-8)\,\tilde{b}^2}}{\mathrm{2N}}, \tag{8.1}$$

where squares are computed coordinate by coordinate.
The covariance, intended as the non-diagonal elements of R^Q, is

$$\mathrm{nondiag}\left(\mathrm{R^Q}\right) = \mathrm{nondiag}\left(\frac{1}{\mathrm{2N}}\left[\tilde{\mathrm{b}}\tilde{\mathrm{B}}^{\mathrm{T}} + \tilde{\mathrm{B}}\tilde{\mathrm{b}}^{\mathrm{T}} + (\mathrm{2N} - 16)\,\tilde{\mathrm{b}}\tilde{\mathrm{b}}^{\mathrm{T}}\right]\right), \qquad (8.2)$$

with

$$\tilde{B} = \tilde{A}_1 + \ldots + \tilde{A}_4 + \tilde{a}_1 + \ldots + \tilde{a}_4.$$

Then, putting together equations (8.1) and (8.2),

$$R^Q = D + L^R \left[\frac{1}{2N}\begin{pmatrix} 2N - 16 & 1 \\ 1 & 0 \end{pmatrix}\right] L^{R^T},$$

where

$$L^R = \left(\begin{array}{c|c} \tilde{b} & \tilde{B} \end{array}\right)$$

and D is a diagonal matrix such that

$$\mathrm{diag}\left(\mathrm{D}\right) = \frac{\tilde{\mathrm{A}}_1^2 + \ldots + \tilde{\mathrm{A}}_4^2 + \tilde{\mathrm{a}}_1^2 + \ldots + \tilde{\mathrm{a}}_4^2 + 8\tilde{\mathrm{b}}^2 - 2\left(\tilde{\mathrm{b}} * \tilde{\mathrm{B}}\right)}{\mathrm{2N}},$$

with $*$ to indicate the product coordinate by coordinate.

8.3 Assimilation results

As shown on Figure 8.2, the system reproduces the seasonal dynamics of surface chlorophyll concentration, with a maximum during winter and a minimum during summer.
The Data Assimilation scheme provides a closer evolution to the satellite if compared with the control run without assimilation.
The skill of the method is assessed in terms of RMSD of the difference between satellite data and forecast. Figure 8.3 shows the RMSD evolution in time.
Finally, Figures 8.5 and 8.4 show that the SEIK does not degenerate the other variable, in particular nitrate and phosphate. The implemented method behaves as expected, leading to a better estimation of the state of the system.

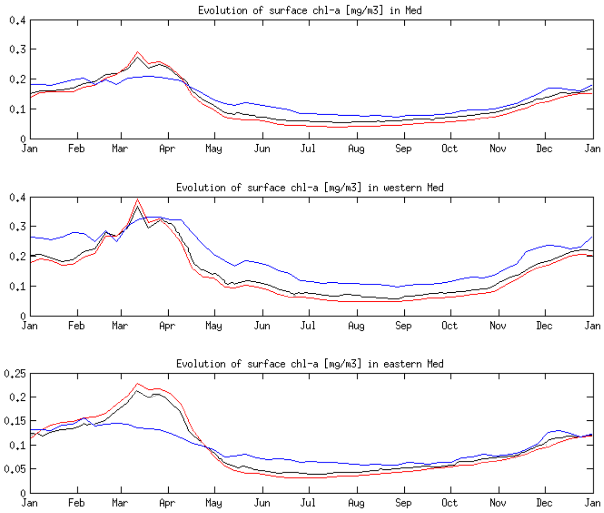

Figure 8.2: Surface chlorophyll concentration. Black: modified SEIK, red: control run, blue: satellite

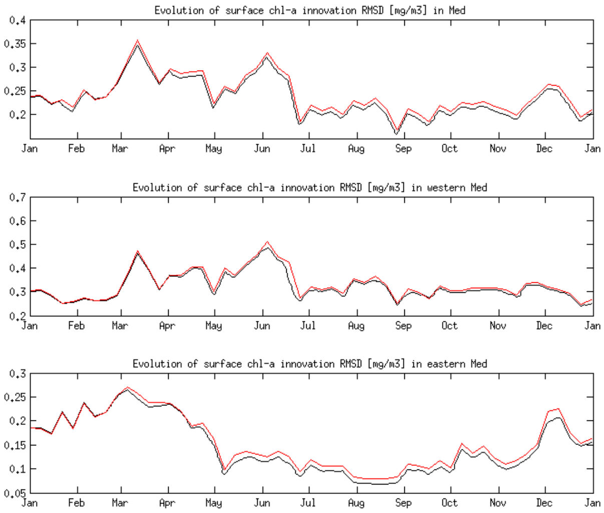

Figure 8.3: Evolution of RMSD. Black: modified SEIK, red: control run

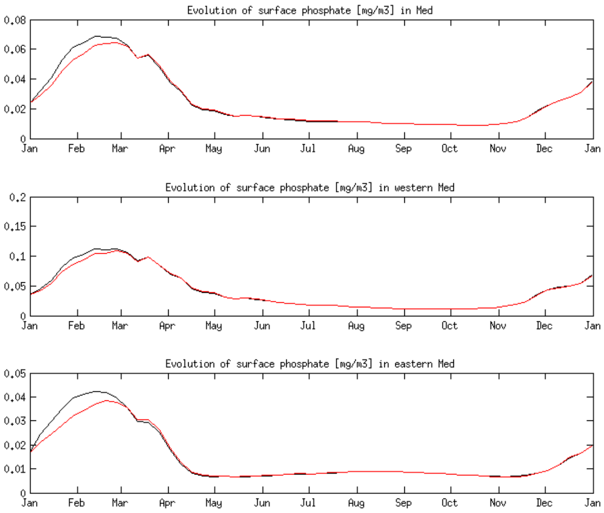

Figure 8.4: Evolution of phosphate at surface. Black: modified SEIK, red: control run

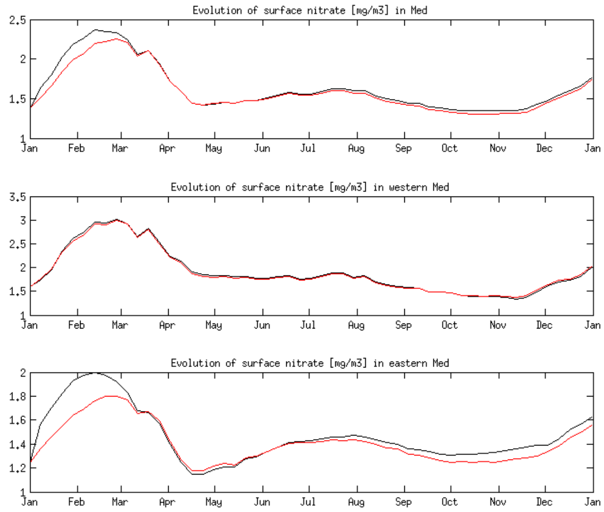

Figure 8.5: Evolution of nitrate at surface. Black: modified SEIK, red: control run

8.4 Parallelization

The OGSTM-BFM is coded in fortran, using MPI for the parallelization. At a resolution of $\frac{1}{4}^o$ when running with 102 cores, it simulates 1 year in around 40 minutes plus 5 minutes of initialization.

Each process is responsible for an area of the Mediterranean Sea, and communicates with its neighbours to exchange information at boundaries.

In order to implement the Data Assimilation scheme, new layers of processes has been added, one for each ensemble member. Each process became responsible for one area in one ensemble member, and communicates with its neighbours, as well with other ensemble members processes operating in the same area.

Figure 8.6 pictures the parallelization structure. There are no connections along the diagonals, in fact due to the structure of the SEIK equations, it is possible to leave the matrices distributed along the processes (each one in its area), and use vertical connections to solve the computations area by area. This property is very useful to reduce communication time and rise the efficiency.

The equations in Section 5.2 are probably the most difficult to implement without using diagonal connections. The result has been achieved by transforming equation (5.54), i.e.

$$R_i^l = R_i + R_i^Q - \hat{Y}^f W \hat{Y}^{fT} + \hat{y}_i^f \hat{y}_i^{fT} + \hat{y}_i^f {y'_i}^T + y'_i \hat{y}_i^{fT} - \left(y_i^f {y'_i}^T + y'_i {y_i^f}^T \right),$$

by splitting it in a diagonal part and a low rank part, namely

$$R_i^l = \Lambda + C \Gamma C^T,$$

where Λ is a diagonal matrix,

$$\Lambda = R_i + D,$$

and Γ is a small matrix computed knowing that

$$C = \left(\begin{array}{c|c|c|c|c|c} Y^f & \hat{y}_i^f & y_i^f & y'_i & \tilde{b} & \tilde{B} \end{array} \right).$$

In this way, finally, the inverse is calculated accordingly to the formula

$${R_i^l}^{-1} = \Lambda^{-1} - \Lambda^{-1} \left(C^T \Lambda C + \Gamma \right) \Lambda^{-1}.$$

The computation has been done using 1326 cores, and has been completed in around 70 minutes, plus 40 of initialization.

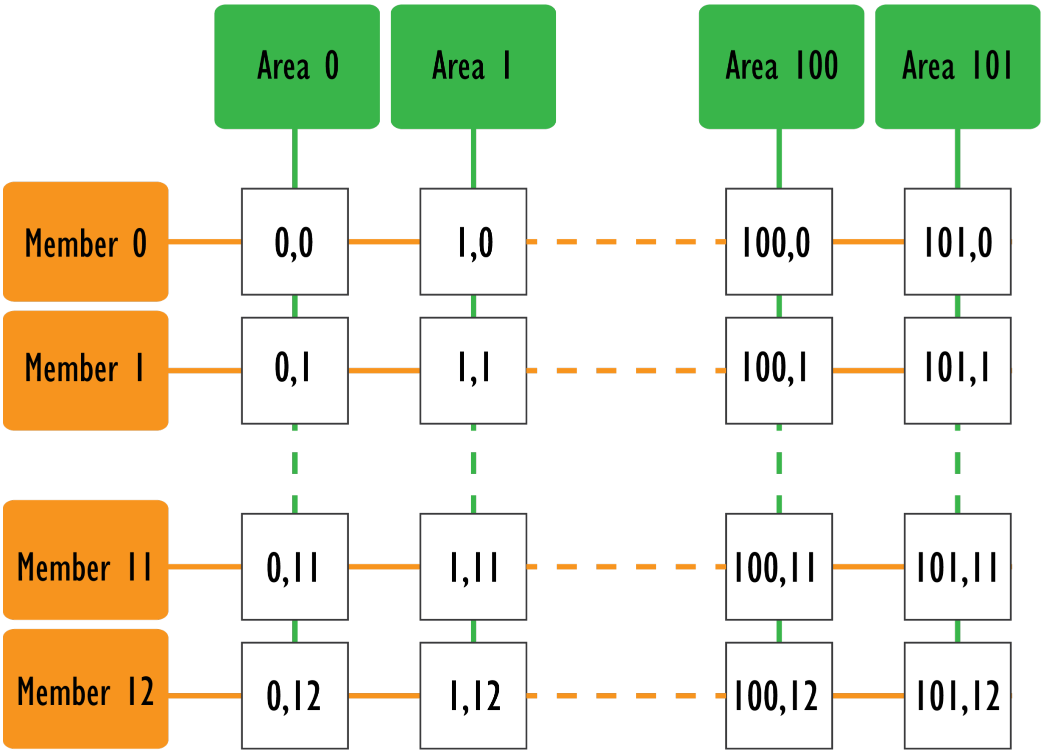

Figure 8.6: Parallel structure of implementation

Chapter 9

Conclusions and future work

Data Assimilation is a field in fast expansion, in particular when referred to geosciences. In the context of marine biogeochemistry, this work aimed to two objectives.

The first one was the comparison between two Data Assimilation methods, namely 3D-VAR and SEIK, the former using a variational approach, the latter based on the Kalman-Filter.

Using a Bayesian framework, Chapter 3 has shown that the two approaches have various similarities, like the fact that both derive from the same Gaussian model and observation error hypothesis. The main differences instead are:

- the choice of the state estimator, that is the mode for the variational method and the mean for the Kalman-filter,
- Kalman-filter's capability to transfer covariance from one time step to the next one, while 3D-VAR simply neglects it,
- Kalman-filter's necessity of linear operators.

The first difference is not necessary a 3D-VAR weakness but it is at least a debatable choice.

The second one instead has clearly a huge impact on the estimation, and represents one of the main strength points of the Kalman-Filter approach.

The third one seems a very limiting Kalman-Filter weakness, but the problem is completely resolved in Chapter 4 with the introduction of the SEIK filter.

This means that, under a theoretical point of view, the SEIK filter is superior to the 3D-VAR (that, on the other hand, is cheaper in terms of computational cost).

The experiment in Chapter 6 empirically confirms the theoretical results. In fact, the 3D-VAR algorithm, applied to a Fasham-like biogeochemical model coupled to a physical transport model, produced errors one order of magnitude higher than SEIK, in terms of RMSD.

The second objective of this thesis was the development of a novel filter based on SEIK, suited for biogeochemical marine systems, with a specific focus on the effects and the estimation of the model error.

At this purpose, the SEIK algorithm has been generalised (Section 4.5) and modified (Chapter 5). In particular, in Sections 5.1, a new sampling technique has been added, which is able to match higher orders approximations of the operators, obtaining more precise numerical calculations of the mean and covariance. Furthermore, the full model error has been taken into account (Section 5.2), by splitting it into two components. The first one is interpreted as model uncertainty, the other one is considered as an added measurement error induced by a noise-like effect. In Section 5.3, a maximum likelihood strategy has been proposed for the estimation of the model error covariance. Lastly, an *ad hoc* smoother has been developed and presented in Section 7.1, with the remarkable feature of a very low computational cost.

Chapter 6 tests proved that SEIK's modified version has the same skill as standard SEIK when the model error is small or the ensemble size is big enough. On the contrary, in the case of large model error and small ensemble size, the modified SEIK provides an up to 10% better RMSD compared to the standard SEIK, showing an improved resilience to divergences induced by model errors.

The tests on the maximum likelihood algorithm obtained estimations comparable to the true values, with the tendency to overestimate in both filters. However, as discussed in Section 6.8, the results obtained with the modified version of the SEIK algorithm were much more accurate (even up to one order of magnitude) than the standard SEIK's results. It is interesting to note that, in all the tested cases, the RMSD error of the filter was slightly better if the estimated model error was used in place of the true value. A possible explanation is that the true model error value is actually an underestimation, in fact it does not account for the Data Assimilation scheme internal errors and approximations.

The tests on the smoother, in Section 7.2, show a clear improvement in the quality of the estimation of the state of the system, sensibly augmenting the performance of the assimilation, with even better results on large model error experiments.

Finally, Chapter 8 demonstrates the feasibility of the implementation of the novel Data Assimilation scheme for a realistic 3D biogeochemical marine model. Further, the chapter provides an efficient computational solution to face the parallelization issue (Section 8.4). Even if the results are not extensively discussed in Section 8.3, the 1-year 3D test clearly shows a behavior that respects the seasonal dynamics without degenerating the not assimilated variables. The 3D implementation highlighted the importance of a tuning phase of some elements of the scheme settings before the biogeochemical results can be discussed into details. For example, the logarithm of the concentrations of very diluted tracers can grow too fast, producing instabilities, and a careful setting of cutting thresholds is necessary.

To fully exploit the potentiality of the novelty of the thesis some future works are foreseen.

Firstly, tuning and refinement of the 3D realistic implementation will allow a punctual analysis of the biogeochemical results in the Mediterranean Sea. Moreover, it would be interesting to reproduce, on other Kalman-based filters, the same improvement strategies used for SEIK. In particular, the high order exact sampling (Section 5.1) is very general in its development, and can probably lead to good results in various ensemble-based filters.

Bibliography

[1] *The Biogeochemical Flux Model (BFM), Equation Description and User Manual, BFM version 5.1, Release 1.1*, August 2015.

[2] Herv Abdi and Lynne J. Williams. Principal component analysis, 2010.

[3] J.L. Anderson. An adaptive covariance inflation error correction algorithm for ensemble filters. *Tellus*, 59A, 2007.

[4] A.F. Bennet. Inverse modeling of the ocean and atmosphere.

[5] A.F. Bennet. Inverse methods in physical oceanography. *Cambridge University Press*, 2016.

[6] P. Brasseur, N. Gruber, R. Barciela, K. Brander, M. Doron, A. El Moussaoui, A.J. Hobday, M. Huret, A.-S. Kremeur, P. Lehodey, R. Matear, C. Moulin, R. Murtugudde, I. Senina, and E. Svendsen. Integrating biogeochemistry and ecology into ocean data assimilation systems. *Oceanography*, 22 (3), 2009.

[7] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wires climate change*, 9, 2018.

[8] G. Cossarini, P. Lazzari, and C. Solidoro. Spatiotemporal variability of alkalinity in the mediterranean sea. *Biogeosciences*, 12(6), 2015.

[9] G. Cossarini, P.F.J. Lermusiaux, and C Solidoro. Lagoon of venice ecosystem: Seasonal dynamics and environmental guidance with uncertainty analyses and error subspace data assimilation. *Journal of Geophysical Research*, 114, 2009.

[10] R. Van der Merwe and E. A. Wan. The square-root unscented kalman filter for state and parameter-estimation. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, volume 6, pages 3461–3464, 2001.

[11] S. Dobricic and N. Pinardi. An oceanographic three-dimensional variational data assimilation scheme. *Ocean Modelling*, 2008.

[12] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[13] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53, 2003.

[14] H. Fang, N. Tian, Y. Wang, M. Zhou, and M. A. Haile. Nonlinear bayesian estimation: from kalman filtering to a broader horizon. *IEEE/CAA Journal of Automatica Sinica*, 5(2):401–417, Mar 2018.

[15] M. J. R. Fasham, H. W. Ducklow, and S. M. McKelive. A nitrogen-based model of plankton dynamics in the oceanic mixed layer. *Journal of Marine Research*, 48, 1990.

[16] P. A. Gagniuc. Markov chains: From theory to implementation and experimentation, 2017.

[17] N. Halko, P. G. Matinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53(2):217–288, 2011.

[18] S. Haykin. Kalman filtering and neural networks. 2001.

[19] J.S. Hesthaven, G. Rozza, and B. Stamm. Certified reduced basis methods for parametrized partial differential equations. 2015.

[20] I. Hoteit, D. Pham, and J. Blum. A simplified reduced order kalman filtering and application to altimetric data assimilation in tropical pacific. *Journal of Marine Systems*, 36, 2002.

[21] K. Ide, P. Courtier, M. Ghil, and A.C. Lorenc. Unified notation for data assimilation: Operational, sequential and variational. *Data Assimilation in Meteorology and Oceanography: Theory and Practice*, 75, 1997.

[22] T. Janjic, L. Nerger, A. Albertella, J. Schroter, and S. Skachko. On domain localization in ensemble-based kalman filter algorithms. *Monthly Weather Review*, 139.

[23] A. H. Jazwinski. Stochastic processes and filtering theory, 1970.

[24] S. Juler and J. Uhlmann. A new extension of the kalman filter to nonlinear systems. *Storage and Retrieval for Image and Video Databases.*

[25] R. E. Kalman. A new approach to linear filtering and prediction problems. *Transaction of the ASME - Journal of Basic Engineering*, pages 35–45, 1960.

[26] K.J.H. Law, A.M. Stuart, and K.C. Zygalakis. Data assimilation: A mathematical introduction. 2015.

[27] P. Lazzari, C. Solidoro, V. Ibello, S. Salon, A. Teruzzi, K. Béranger, S. Colella, and A. Crise. Seasonal and inter-annual variability of plankton chlorophyll and primary production in the mediterranean sea: a modelling approach. *Biogeosciences*, 9, 2012.

[28] P. Lazzari, C. Solidoro, S. Salon, and G. Bolzon. Spatial variability of phosphate and nitrate in the mediterranean sea: A modeling approach. *Deep-Sea Research I*, 108, 2016.

[29] P. Lazzari, A. Teruzzi, S.a Salon, S. Campagna, C. Calonaci, S. Colella, M. Tonani, and A. Crise. Pre-operational short-term forecasts for the mediterranean sea biogeochemistry. *Ocean Science*, 6, 2010.

[30] L. Ljung. Asymptotic behavior of the extended kalman filter as a parameter estimator for linear systems. *IEEE Transactions on Automatic Control*, 24(1):36–50, February 1979.

[31] A.C. Lorenc. Analysis methods for numerical weather prediction. *Quart*, 112, 1986.

[32] A.C. Lorenc. Novel approach to nonlinear/non-gaussian bayesian state estimation. *IEEE Proceedings-F*, pages 107–113, 1993.

[33] D. Melaku Canu, A. Ghermandi, P.A.L.D. Nunes, P. Lazzari, G. Cossarini, and C. Solidoro. Estimating the value of carbon sequestration ecosystem services in the mediterranean sea: An ecological economics approach. *Global Environmental Change*, 32, 2015.

[34] C. Montella. The kalman filter and related algorithms: A literature review. 05 2011.

[35] H. Moradkhani and K.-L. Hsu. Uncertainty assessment of hydrologic model states and parameters: Sequential data assimilation using the particle filter. *Water Resources Research*, 41, 2005.

[36] L. Nerger, T. Janjic Pfander, W. Hiller, and J. Schrter. Seik - the unknown ensemble kalman filter. In *5th WMO Symposium on Data Assimilation, Melbourne, Australia, October 5 - 9*, 2009.

[37] D. T. Pham. A singular evolutive interpolated kalman filter for data assimilation in oceanography, 1996.

[38] D. T. Pham, J. Verron, and M. C. Roubaud. A singular evolutive extended kalman filter for data assimilation in oceanography. *Journal of Marine Systems*, 16, 1998.

[39] K. Ponomareva, P. Date, and Z. Wang. A new unscented kalman filter with higher order moment-matching, 2010.

[40] S. Salon, G. Cossarini, P. Lazzari, A. Teruzzi, A. Crise, and C. Solidoro. Reanalysis of mediterranean sea biogeochemistry and the quest for biogeochemical seasonal forecasts.

[41] C. Schillings and A.M. Stuart. Analysis of the ensemble kalman filter for inverse problems, 2016.

[42] A.M. Stuart. The bayesian approach to inverse problems.

[43] D Tenne and T. Singh. The higher order unscented filter. In *Proceedings of the 2003 American Control Conference, 2003.*, volume 3, pages 2441–2446, 2003.

[44] A. Teruzzi, S. Dobricic, C. Solidoro, and G. Cossarini. A 3-d variational assimilation scheme in coupled transport-biogeochemical models: Forecast of mediterranean biogeochemical properties. *J. Geophys. Res. Oceans*, 119, 2014.

[45] Anna Teruzzi, Giorgio Bolzon, Stefano Salon, Paolo Lazzari, Cosimo Solidoro, and Gianpiero Cossarini. Assimilation of coastal and open sea biogeochemical data to improve phytoplankton simulation in the mediterranean sea. *Ocean Modelling*, 132:46 – 60, 2018.

[46] M.K. Tippet, J.L. Anderson, C.H. Bishop, T.M. Hamill, and J.S. Whitaker. Analysis methods for numerical weather prediction. 2002.

[47] G. Triantafyllou, I. Hoteitb, and G. Petihakisa. A singular evolutive interpolated kalman filter for efficient data assimilation in a 3-d complex physical-biogeochemical model of the cretan sea. *Journal of Marine Systems*, 40–41, 2003.

[48] J Tdter and B. Ahrens. A second-order exact ensemble square root filter for nonlinear data assimilation. *Tdter and Ahrens*, 140, 2015.

[49] E. A. Wan and R. Van Der Merwe. The unscented kalman filter for nonlinear estimation. In *Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, pages 153–158, 2000.

[50] A.T. Weaver, J. Vialard, and D.L.T. Anderson. Three- and four-dimensional variational assimilation with a general circulation model of the tropical pacific ocean. part i: Formulation, internal diagnostics, and consistency checks. *Monthly Weather Review*, 131, 2003.

[51] J.S. Whitaker and T.M. Hamill. Evaluating methods to account for system errors in ensemble data assimilation. *Monthly Weather Review*, 140, 2012.