Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Delft Institute of Applied Mathematics

Improving the Algal Bloom Prediction in the North Sea by Ensemble Kalman Filtering in the GEM/BLOOM Model

A thesis submitted to the
Delft Institute of Applied Mathematics
in partial fulfillment of the requirements

for the degree

MASTER OF SCIENCE
in
APPLIED MATHEMATICS

by

Elisabeth Geraldine Rens

Delft, the Netherlands
August 2013

TU Delft

MSc THESIS APPLIED MATHEMATICS

"Improving the Algal Bloom Prediction in the North Sea by Ensemble Kalman Filtering in the GEM/BLOOM Model"

Elisabeth Geraldine Rens

Delft University of Technology

Daily supervisor

G.Y.H. El Serafy PHD

Responsible professor

Prof.dr.ir A.W. Heemink

Other thesis committee members

Prof.dr.ir C. Vuik

August 2013

Delft, the Netherlands

Abstract

The ecological state of the North Sea surface water can be indicated by ocean variables such as the Chlorophyll-a (*Chlfa*) concentration. *Chlfa* is the principal photosynthetic pigment and is common to all phytoplankton and can therefore be used as a measure of phytoplankton biomass. The GEM/BLOOM model developed at Deltares is a generic ecological model that simulates transport of substances in a water system along with various ecological processes. This model is able to estimate the *Chlfa* concentration. Models are always prone to errors due to assumptions made in the development and the use of numerical approximations. Such errors can be reduced through the use of data assimilation and thus can significantly improve the forecast. The ensemble Kalman filter (EnKF) is a generic data assimilation method which is suited for highly nonlinear models with a large scale. This filter is validated by the use of twin experiments on the GEM/BLOOM model. It successfully improves the prediction of *Chlfa*, but however shows filter divergence in some grid points. The performance is further improved by the use of the Ensemble Square Root Filter (ESRF) with a localized analysis. Finally, application of this filter to assimilating daily MERIS remote sensing images is explored and shows to be promising, but requires more tuning before it can operate.

Acknowledgments

The work for this thesis was done at Deltares, an institute for applied research in the field of water, subsurface and infrastructure. The thesis was part of the COBIOS project that aims to integrate satellite products and ecological models into an operational information service on high biomass blooms in Europes coastal waters.

It was a great opportunity and a pleasure to undertake a project at Deltares in the second year of my master's study. I would like to thank Ghada El Serafy and Arnold Heemink for their support and useful feedback. Also, a thank you to Sandra Gaytan for her technical support. Furthermore, I would like to express my gratitude to Asia Pelc and Martin Verlaan for their advice on data assimilation. Also, a thank you to Peter Wilders, for discussing my report. Kees Vuik, I think for being in the thesis committee. I also thank the other students at Deltares who made my stay very pleasant. Finally, a thank you to my boyfriend, all my family and friends, for greatly supporting me, but also for making sure that I made time to relax.

Contents

1 Introduction

Good water quality is vital to the world and its inhabitants, as oceans, rivers and coastal waters have a big impact on our lives and environment. Effects range from the intake of food retrieved from the ocean to the environmental effects of nutrient discharge. An important aspect of water quality research is comprehending the processes that are involved. This understanding can provide an improved prediction of the state of water quality, which in turn enables the study of management choices.

Phytoplankton is an important indicator for water quality [30], as it affects many factors related to the ecological quality of the water, such as turbidity. Harmful algal blooms increasingly cause severe economic and ecological damage in both marine and freshwater systems worldwide [18]. Excessive phytoplankton blooms are generally caused by high nutrient levels [30]. These nutrients are greatly released into the environment as a result of human activities. Algal blooms can result in deoxygenation of the water mass when large masses of algae die and decompose, leading to the death of aquatic plants and animals. Not only is this a danger to biodiversity, but it also causes mass mortality in fish and mussel farms [25]. Algal blooms can be harmful to the community in many other ways. Some effects are bad odors, lack of oxygen, blocking filters in water transportation systems and damage of recreation [30].

Phytoplankton blooms mainly occur in late spring or summer in the Dutch coastal zone, when environmental conditions for phytoplankton growth are optimal [24]. Algal blooms became of particular concern along the Dutch North Sea coast following the construction of the Delta Works after the 1953 flood event [18]. Due to growing population density, it is predicted that inflows from nutrient loaded rivers into the Dutch North Sea will increase and subsequently will lead to more algal bloom events [18]. Furthermore, it is expected that the seasonality and spatial distribution of major bloom forming algae will change [9].

For these reasons, it is of particular interest to increase our understanding of the processes that affect algal bloom growth. Moreover, there is a need for a reliable prediction of algal bloom development, as this may provide a warning system for involved parties and the public. Another interest is to be able to investigate the effect of management choices for potential damage reduction. For the aforementioned purposes, Deltares developed a model called GEM/BLOOM. This system is used to make predictions and evaluate management policies concerning water quality and phytoplankton blooms in the North Sea. It has been validated in the North Sea and has proven to be a valuable tool [21].

2 Problem description

The ecological state of surface water can be indicated by ocean variables such as the concentration of Chlorophyll-a (*Chlfa*), the Diffuse Attenuation Coefficient (K_d), Suspended Particulate Matter (*SPM*), Colored Dissolved Organic Matter (*CDOM*), and more [18]. The model GEM/BLOOM [19] is able to estimate these variables. Alternatively, satellite images provide large scale measurement maps at high spatial and temporal resolution for *Chlfa* and K_d. However, the observations lack complete coverage due to cloud interference and lack predictive power [27]. Furthermore, the fact that the variables are not actually observed, but derived from ocean color images [3], results in more uncertainties. Obviously, the model is also imperfect, due to several factors. For instance, errors arise due to assumptions that are made in the development of the model. Also, errors are made by the use of numerical solvers. In addition, the parameters that are estimated by experience might be faulty.

Integrating the remote sensing products with the model is a possible solution to this problem, as it can provide a more accurate estimate of the state of the observed system [3]. This model-data integration can be accomplished by Data Assimilation, which is a powerful tool to merge imperfect dynamical models and uncertain observations. Several methods for Data Assimilation are available. These type of algorithms combine model forecasts and available measurements in an optimal way, i.e., they intend to minimize the combined uncertainties of model and measurements. This master thesis focuses on ensemble based filtering methods. Ensemble based filtering methods have some appealing properties, which make them suitable for the type of model in this thesis. First of all, these type of methods are generic and applicable to complex and nonlinear models, like the model at hand, no linearization of the model is needed. Secondly, ensemble based filters allow us to work around the model, use it as a black box. Consequently, it is not required to adapt the algorithm when a new version of the model is available, which occurs quite frequently. Furthermore, ensemble based filters are easily extended. Also, the cluster available at Deltares allows us to apply parallel computation, this can be exploited by using ensemble based methods to decrease computational burdens. Finally, at Deltares the use of ensemble based filtering has been investigated with this and other models and has shown to be very promising [35], [3].

3 Thesis outline

This thesis gives a brief description of the GEM/BLOOM, in chapter four. In this chapter, also the North Sea, the area of interest, is introduced. Furthermore, the outputs *Chlfa* and K_d are described. Then a description of the MERIS data and *in situ* measurements followed by an explanation in what manner they will be used in chapter five. Next, the methodology of the data assimilation is explained in chapter six. In this chapter, two ensembles based filters are described, namely the Ensemble Kalman Filter (EnKF) and the Ensemble Square Root Filter (ESRF). In addition, a validation method for the filters, by means of a twin experiment, is proposed. An important feature of Data Assimilation is the system noise. The way in which this system noise is modelled influences the filter performance greatly. When the modelled system noise describes the uncertainty well, the filter will perform better. Thus, in chapter seven, two ways of modelling this system noise are considered. Next, a way of further improving the filter, by means of covariance localization is described in chapter eight. In chapter nine, the propagation of the system noise by the model is investigated, so that a measurement noise can be defined for the twin experiments. Furthermore, these results can also be used to define the system noise for the real data application of the filter.

Chapter ten deals with choosing a filter by performing twin experiments. In this chapter the two ensemble based filters, are compared and a system noise is chosen. Furthermore, it is investigated if covariance localization is desired. Finally, with the ultimate chosen filter, it is tested how large the ensemble size can be chosen. In figure 1 the structure of chapter nine is shown. The figure also shows that after choosing a filter and ensemble size, the filter will be applied to the GEM/BLOOM model with real data, MERIS data. This is done in chapter eleven. Then this thesis ends with a discussion, conclusion and recommendations in chapters twelve, thirteen and fourteen respectively. References and the appendix can be found in chapter fifteen and sixteen. The appendix consists of additional figures, a list of figures and tables.

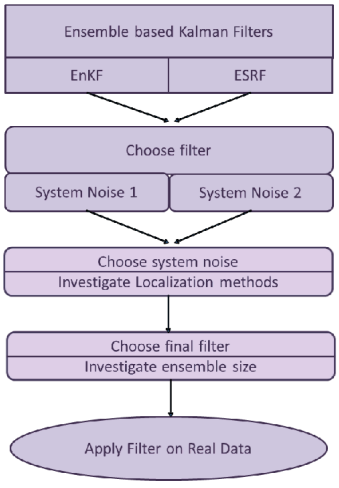


Figure 1: ***Research structure***

4 Model description

This chapter gives an overview of the GEM/BLOOM model. As the model is of high complexity and involves a large variety of processes, only a global overview is given. Important features, such as the state variables and relevant parameters are highlighted. The GEM/BLOOM is able to calculate, among others, phytoplankton compositions, primary production, colored dissolved organic matter, a diffuse attenuation coefficient, and chlorophyll-a concentration. This thesis focuses on assimilating and improving the prediction of *Chlfa* and K_d, as those are the most important water quality indicators with respect to phytoplankton plus *in situ* and MERIS remote sensing observations are available. Therefore, a short description of the *Chlfa* and K_d are given in this chapter, followed by a description of the North Sea, which is the area of interest.

The GEM/BLOOM model [4] is a generic ecological model that simulates transport of substances in a water system along with various ecological processes. It is able to calculate, among others, phytoplankton compositions, primary production, a diffuse attenuation coefficient, and chlorophyll-a concentration.

GEM/BLOOM simulates the nutrient cycles of nitrogen (N), phosphorus (P) and silicate (Si). A finite grid approach is used to solve the advection-diffusion equation that models the transport of substances. Processes in the water are included by sources and sinks in the advection-diffusion equation. This amounts to the following equation

$$\begin{aligned} \frac{\partial C}{\partial t} &= -u\frac{\partial C}{\partial x} - v\frac{\partial C}{\partial y} - w\frac{\partial C}{\partial z} \\ &\quad + \frac{\partial}{\partial x}(D_x\frac{\partial C}{\partial x}) + \frac{\partial}{\partial y}(D_y\frac{\partial C}{\partial y}) + \frac{\partial}{\partial z}(D_z\frac{\partial C}{\partial z}) \\ &\quad + S + P. \end{aligned} \tag{1}$$

where

C : concentration (gm^{-3})

u, v, w : components of the velocity vector (ms^{-1})

D_x, D_y, D_z : components of the dispersion tensor (m^2s^{-1})

x, y, z : components in three spatial dimensions (m)

S : sources and sinks of mass due to loads and boundaries

P : sources and sinks of mass due to processes

t : time (s)

A schematic overview of the processes and variables that are incorporated in the model is shown in figure 2. The processes affect the state variables in the model by making them appear, disappear or change into another state. The

phytoplankton growth process is modelled by a module in GEM/BLOOM that is called BLOOM [19], for which a description follows.

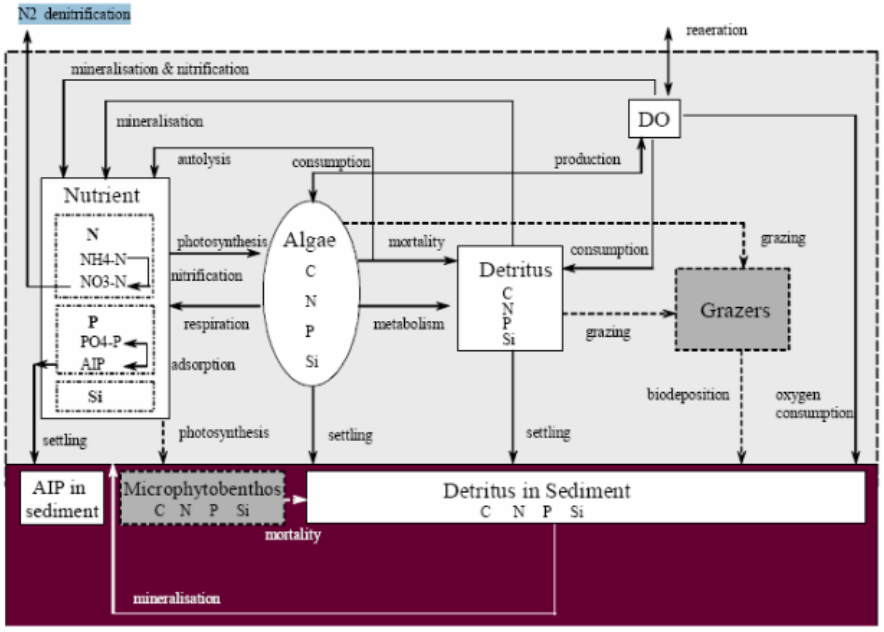


Figure 2: ***A schematic view of the GEM/BLOOM model [4]***

4.1 BLOOM

In the BLOOM module, four phytoplankton species groups are simulated: diatoms, flagellates, dinoflagellates and *Phaeocystis*. Within each of these groups, three phenotypes are defined to account for adaptation to changing environmental conditions:

energy types : have relatively high growth rates, low mortality rates and high N/C and P/C ratio

nitrogen types : have typically lower internal N/C ratio, lower maximum growth rates, higher mortality rates, higher settling velocities and higher chlorophyll content

phosphorus types : have typically lower internal P/C ratio, lower maximum growth rates, higher mortality rates, lower settling velocities and lower chlorophyll-a content

The different phenotypes of a species group are modelled as separate variables with different parameter settings, e.g. growth rates, settling velocities and respiration rates and factors for converting biomass to chlorophyll-a concentrations.

BLOOM is based upon the principle of competition between different species and mathematically this comes down to optimizing the total net production of

the phytoplankton community under certain constraints. Thus, the BLOOM module calculates the growth of algae as a function of nutrient and light conditions. For the optimization, a linear programming technique is used. The constraints that limit the phytoplankton growth are a growth constraint, mortality constraint, light constraint and nutrient uptake constraints. The nutrient uptake constraint is defined by the available resources calculated by GEM.

A mathematical description of this module is to maximize

$$\sum_{i=1}^{n}(pg_i \cdot le_i - r_i) \cdot ALG_{i,new}, \tag{2}$$

such that we satisfy a growth constraint per species group

$$\sum_{i=1}^{3} ALG_{i,new} \leq \sum_{i=1}^{3} ALG_i \cdot e^{(pg_i \cdot le_i - r_i)\cdot \Delta t}, \tag{3}$$

a mortality constraint per species group

$$\sum_{i=1}^{3} ALG_{i,new} \geq \sum_{i=1}^{3} ALG_i \cdot e^{-m_i \cdot \Delta t}, \tag{4}$$

a nitrogen constraint

$$\sum_{i=1}^{3} S_{N,i} \cdot ALG_{i,new} \leq \sum_{i=1}^{3} S_{N,i} \cdot ALG_i + NO_3 + NH_4, \tag{5}$$

a phosphorus constraint

$$\sum_{i=1}^{3} S_{P,i} \cdot ALG_{i,new} \leq \sum_{i=1}^{3} S_{P,i} \cdot ALG_i + PO_4, \tag{6}$$

a silicate constraint

$$\sum_{i=1}^{3} S_{Si,i} \cdot ALG_{i,new} \leq \sum_{i=1}^{3} S_{Si,i} \cdot ALG_i + Si, \tag{7}$$

and a light constraint

$$K_{min,i} \leq K_d \leq K_{max,i}, \tag{8}$$

where pg_i represents the maximal gross growth rate algae type i, le_i represents the growth efficiency of algae type i, r_i represents the maintenance respiration rate for algae type $i$and m_i represents the mortality rate for algae type i. $S_{Nut,I}$, represent the nutrient to carbon ratios of each algae type. Finally, K_d, represents the total extinction coefficient.

4.2 Key aspects of the model

In this section the most important features of the model are described. In total 26 [4] state variables and 422 [30] parameters are used in the model. The state variables are listed in table 1. The state variables can be divided into active and inactive substances. The ones that are buried in the sediment are called inactive substances. All other state variables are called active substances, they are present in the whole of the sea. The state variables and parameters are used to calculate indicators for water quality. This master thesis focuses on the improvement of the prediction of the variables *Chlfa* and K_d. The relevant parameters belonging to these variables can be found in table 2. Note that in the model some of the parameters are constrained, by for instance, the light constraint in the BLOOM module, equation 8. A complete list of the parameters as well as its detailed description can be found in [4]. These parameters and some of these state variables are used in order to calculate the *Chlfa* and K_d, for which the descriptions now follows.

Symbol	**Description**	**Unit**
NO_3	nitrate	$gN \cdot m^{-3}$
NH_4	ammonium	$gN \cdot m^{-3}$
PO_4	ortho-phosphate	$gP \cdot m^{-3}$
Si	dissolved silicate	$gSi \cdot m^{-3}$
O_2	dissolved oxygen	$gO_2 \cdot m^{-3}$
ALG_1	diatoms energy type	$gC \cdot m^{-3}$
ALG_2	diatoms nitrogen type	$gC \cdot m^{-3}$
ALG_3	diatoms phosphorus type	$gC \cdot m^{-3}$
ALG_4	flagellates energy type	$gC \cdot m^{-3}$
ALG_5	flagellates nitrogen type	$gC \cdot m^{-3}$
ALG_6	flagellates phosphorus type	$gC \cdot m^{-3}$
ALG_7	dinoflagellates energy type	$gC \cdot m^{-3}$
ALG_8	dinoflagellates nitrogen type	$gC \cdot m^{-3}$
ALG_9	dinoflagellates phosphors type	$gC \cdot m^{-3}$
ALG_{10}	*Phaeocystis* energy type	$gC \cdot m^{-3}$
ALG_{11}	*Phaeocystis* nitrogen type	$gC \cdot m^{-3}$
ALG_{12}	*Phaeocystis* phosphorus type	$gC \cdot m^{-3}$
POC	particulate organic carbon	$gC \cdot m^{-3}$
PON	particulate organic nitrogen	$gN \cdot m^{-3}$
POP	particulate organic phosphorus	$gP \cdot m^{-3}$
$POSi$	particulate organic silicate	$gSi \cdot m^{-3}$
POC_s	particulate organic carbon in the sediment	$gC \cdot m^{-3}$
PON_s	particulate organic nitrogen in the sediment	$gN \cdot m^{-3}$
POP_s	particulate organic phosphorus in the sediment	$gP \cdot m^{-3}$
$POSi_s$	particulate organic silicate in the sediment	$gSi \cdot m^{-3}$

Table 1: ***State variables in GEM/BLOOM model***

4.2.1 Chlorophyll-a concentration

Chlorophyll [30] is a light-sensitive pigment that is found in plants, algae and some bacteria. It can absorb light quantums. It is therefore vital for photosynthesis, that is the process of converting light energy into chemical potential

Symbol	Description	Unit
ea_i	specific extinction of the algae type i	m^2gC^{-1}
e_{hum}	extinction due to humic substances in pure freshwater	m^{-1}
e_{spm}	specific extinction of inorganic suspended matter	m^2g^{-1}
e_{poc}	specific extinction of particulate dead organic matter	m^2gC^{-1}
K_b	specific extinction of background	m^{-1}
ca_i	constant for chlorophyll-a over carbon in algae biomass i	

Table 2: ***Relevant parameters for Chlfa and*** K_d

energy, followed by the fixation of inorganic carbon into sugars. Chlorophyll-a (*Chlfa*)[23] is the principal photosynthetic pigment and is common to all phytoplankton. The *Chlfa* concentration can thus be used as a measure of phytoplankton biomass.

Phytoplankton [35], which absorbs and scatters light, can be expressed in chlorophyll-a concentration. As the most important indicator of phytoplankton activity, the higher the *Chlfa* concentration is, the more likely there is an algal bloom. In fact [27], we can usually find a certain *Chlfa* level such that an algal bloom is assumed to occur when the *Chlfa* concentration is greater than that level.

In the model, the chlorophyll-a concentration is computed according to the following formula [30]:

$$Chlfa = 1000 \cdot \sum_{i=1}^{n} ALG_i \cdot ca_i \tag{9}$$

where ca_i is a stoichiometry (ratio of carbon to chlorophyll-a) of *Chlfa* in alga i and ALG_i is the computed biomass of alga i for $i \in \{1, 2, ..., 12\}$. Different parameter values for stoichiometry are used for the different types of algae to accurately simulate the phenomenon that various phytoplankton types have different abilities to adapt to new light or temperature conditions. The specific values can be found in [20]. The multiplication by 1000 is done such that the *Chlfa* concentration has a unit of mg/m^3 in correspondence to the MERIS data.

4.2.2 Diffuse Attenuation Coefficient

The diffuse attenuation coefficient $K_d(490)$ [5] is an indicator of the turbidity of the water column - how visible light in the blue-green part of the spectrum penetrates the water column. It is directly related to the presence of scattering particles in the water column, either organic or inorganic, and thus is an indicator of water clarity.

It's description follows from the Lambert-Beer equation [21], which states that the light intensity at depth z is

$$I(z) = I_0 e^{-K_d z} \tag{10}$$

where I has a unit of Joules $m^{-2}h^{-1}$ and I_0 is the light intensity at the water surface. Now this gives us

$$K_d = \frac{\ln(\frac{I_0}{I(z)})}{z} \tag{11}$$

So K_d has a unit of m^{-1}.

The value of K_d [5] represents the rate at which light at 490 nm is attenuated with depth. For example a K_d of 0.1/meter means that the light intensity will be reduced one natural log within 10 meters of water. Thus, for a K_d of 0.1, one attenuation length is 10 meters. Higher K_d value means lower attenuation depth, and less clarity of water.

In the model K_d [4] is calculated as the sum of extinction due to algae, suspended particulate matter, humic substances from fresh water input, dead particulate organic matter and background extinction:

$$K_d = K_{Chlfa} + K_{hum} + K_{spm} + K_{poc} + K_b. \tag{12}$$

The separate extinction coefficients with corresponding formulas can be found in table 3, the parameters can be found in table 2. SAL_b is the background salinity which has a constant value of 34.97.

Symbol	Description	Formula
K_{Chlfa}	extinction due to algae biomass	$\sum_{i=1}^{n} ea_i \cdot ALG_i$
K_{hum}	extinction due to humic substances from freshwater input	$e_{hum} \cdot (1 - \frac{SAL}{SAL_b})$
K_{spm}	extinction due to suspended particulate matter	$e_{spm} \cdot SPM$
K_{poc}	extinction due to dead particulate organic matter	$e_{poc} \cdot POC$
K_b	background extinction	$0.08m^{-1}$

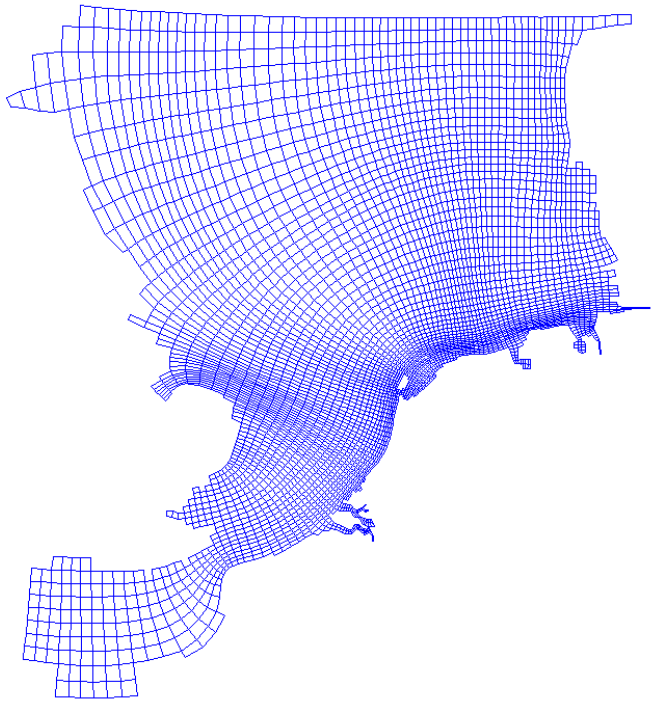
Table 3: ***Extinction coefficients***

4.3 The North Sea

The study area in this research is the North Sea, shown in figure 3a. The North Sea [35] is a marginal sea of the Atlantic Ocean located between Great Britain, Scandinavia, Germany, the Netherlands, and Belgium. An epeiric sea on the European continental shelf, it connects to the ocean through the English Channel in the south and the Norwegian Sea in the north. It is more than 970 kilometers long and 580 kilometers wide, with an area of around 750.000 square kilometers. The North Sea can be characterized as a coastal shelf sea with relatively shallow (10-50 m) coastal waters. Substantial river discharges from Dutch, German, French and UK rivers result in large fluctuations of nutrient concentrations and algal biomass.

The GEM/BLOOM model uses a curvilinear grid [21] of size 134×65, shown in figure 3b. Its horizontal resolution is relatively high in the coastal areas of interest, notably the Dutch coastal zone (approximately 11 km). The grid is much coarser in the northern part of the area included in the model (approximately 20 20 km). This is done so that along the coast, an area of greater interest, the model gives a more precise prediction. The X-Y discretization does not change over time. In the vertical direction, the model consists of 10 layers [35], these can be found in table 4. This discretization does change over time, as the depth differentiates due to the flow model. However, the are consistent with table 4.

(a) Overview of the North Sea with surrounding land



(b) Planar discretization of the North Sea

Figure 3: ***The North Sea***

layer	relative thickness (%)
1(surface)	4.0
2	5.6
3	7.8
4	10.8
5	10.9
6	10.9
7	10.9
8	10.9
9	10.8
10	7.8
11	5.6
12(bottom)	4.0
Total	100.0

Table 4: ***Vertical discretization of the North Sea***

5 Observations

Ocean variables can be measured directly during a sea explorations or derived from remote sensing images, those two methods are described in this chapter. Also, it is explained in what way those observations are used in this thesis. A view of both type of measurements are shown in figure 4.

5.1 MERIS data

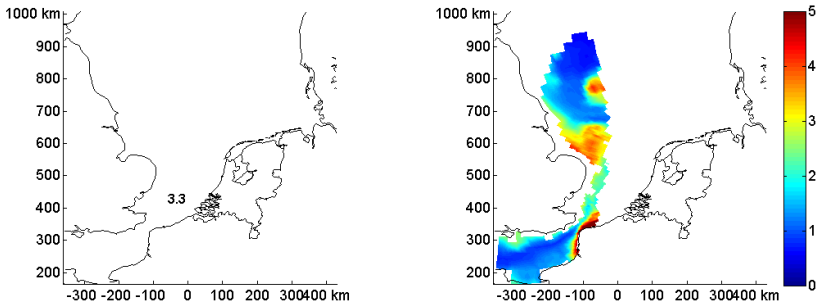
Remote sensing images can frequently provide a spatial view of the North Sea. One of the instruments on board of the ESA's Environmental Satellite Envisat is the Medium Resolution Imaging Spectrometer (MERIS) [27], which is an imaging spectrometer that measures the solar radiation reflected by the Earth and thus provides color images of the oceans. Ocean color images can be converted into a measurement of *Chlfa* and K_d. This is done by the HYDROPT [33] algorithm. Also provided is a value for the uncertainty of the measured variables. The values provided by MERIS and HYDROPT are transformed by Deltares to the North Sea grid.

Remote sensing images offer a powerful tool for providing synoptic observations of the surface of the North Sea frequently. Also, the derived quantities come with a value for the uncertainty. Therefore, these observations are an excellent tool for the Data Assimilation.

5.2 In situ measurements

In situ measurements are observed at the place were the phenomenon occurred. It is possible to measure *Chlfa* concentrations and values for K_d on an expedition on sea. Although the measurement methods have improved over the past 100 years, sampling is still quite limited [36]. Monitoring of phytoplankton [27], which is currently carried out by all North Sea states, is conventionally based on timely and costly water sampling programs. This results in sparse spatial and temporal coverage and gives only a vague impression of the dynamics of phytoplankton.

In this thesis the *in situ* measurements from Rijkswaterstaat [8] and ICES [1] are used. Those *in situ* measurements serve as the true state of the *Chlfa* and K_d, as they are likely to be closer to the true state as other measurements or predictions. Therefore, they are used as a reference for the prediction made by the Data Assimilation.



(a) In situ **(b)** Meris

Figure 4: ***Observations on March 6***

6 Data Assimilation

Data assimilation methods employ imperfect dynamical models and uncertain observations to improve the prediction of the state of a system. One of the most widely applied data assimilation methods is called the Kalman filter. It is an optimal data assimilation method for systems, for which the errors of the model and measurements are assumed to be unbiased and Gaussian. Not only does the filter provide an estimate of the state, but it also gives an estimate of the uncertainty, in agreement with the prescribed error statistics of the model and measurements. The original Kalman Filter was designed for linear dynamical models, but has since evolved into several more complex filters, to make it suitable for large scale non-linear systems.

In the next section of this chapter the Standard Linear Kalman Filter is explained, which provides the general framework of the data assimilation. Since our model is highly non-linear with a large state vector, a description of the Ensemble Kalman Filter follows, which is suitable for this type of model. This filter is a Monte Carlo approach to the original Kalman Filter. This section also covers the adaption of the filter to deal with a non-linear measurement function and shows how parameters can be estimated. Also, a convenient implementation of this filter is given. Then follows a description of the Ensemble Square Root Filter and its implementation, which may show to be an improvement to the Ensemble Kalman Filter. After this, a definition of the system noise is provided. Finally, an explanation of how the filter can be tested by using a synthetic measurement is given. Also explained are the relevant statistics, that can be used to validate the performance of the filter.

6.1 Standard Linear Kalman Filter

In 1960 R.E. Kalman [17] proposed a data assimilation method for linear dynamical systems, known as the Kalman Filter. The objective of the KF is to estimate the state of a discrete time process represented by a linear discrete stochastic model. The basic equations representing the model and the observations read,

$$\mathbf{X}_{k+1} = \mathbf{A}_k \mathbf{X}_k + \mathbf{G}_k \mathbf{W}_k \tag{13}$$

$$\mathbf{Z}_k = \mathbf{H}_k \mathbf{X}_k + \mathbf{V}_k \tag{14}$$

where,

$\mathbf{X}_k$: A vector with the forecast of the system state at time k.

$\mathbf{W}_k$: White system noise, with normal probability distribution, $\mathbf{W}_k \sim N(0, \mathbf{Q}_k)$.

$\mathbf{V}_k$: White measurement noise, with normal probability distribution, $\mathbf{v}_k \sim N(0, \mathbf{R}_k)$.

$\mathbf{A}_k$: Model updating matrix which relates the state at time step k to $k+1$.

$\mathbf{H}_k$: The observation matrix which relates the measurements, $\mathbf{Y}$, to the system state, $\mathbf{X}$, at time step k.

$\mathbf{G}_k$: The matrix which relates the system noise, $\mathbf{W}_k$, to the system state at time step $k, \mathbf{X}_k$.

$\mathbf{Q}_k$: The covariance matrix of the system noise at time step $k, \mathbf{W}_k$.

$\mathbf{R}_k$: The covariance matrix of the measurement noise at time step $k, \mathbf{v}_k$.

The above state space description assumes that the $\mathbf{W}_k$ and $\mathbf{V}_k$ are independent and the covariance matrices, $\mathbf{Q}_k$ and $\mathbf{R}_k$ are positive definite.

The recursive process of the KF can be separated into two parts, the system or time update and the measurement update. In the time update the state of the system is estimated by performing a model update. This results in a forecasted state, which is compared to a measurement. In the measurement update, the measurement is used to improve the estimate of the state. This results in the analyzed state, which is used in the following iteration of the KF. For forecasted variables we will use the notation superscript f and for the analyzed variables superscript a.

The equations for the time update are

$$\mathbf{X}^f_{k+1} = \mathbf{A}_k \mathbf{X}^a_k \tag{15}$$

$$\mathbf{P}^f_{k+1} = \mathbf{A}_k \mathbf{P}^a_k \mathbf{A}^T_k + \mathbf{Q}_k \tag{16}$$

where $\mathbf{P}$ is the covariance matrix of to the forecast error.

The equations for the measurement update are

$$\mathbf{X}^a_{k+1} = \mathbf{X}^f_{k+1} + \mathbf{K}_{k+1}(\mathbf{Y}_{k+1} - \mathbf{H}_{k+1}\mathbf{X}^f_{k+1}) \tag{17}$$

$$\mathbf{K}_{k+1} = \mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1}(\mathbf{H}_{k+1}\mathbf{P}^f_{k+1}\mathbf{H}^T_{k+1} + \mathbf{R}_k)^{-1} \tag{18}$$

$$\mathbf{P}^a_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}^f_{k+1} \tag{19}$$

where $\mathbf{Y}_k$ are the available measurements and $\mathbf{K}_k$ is called the Kalman gain matrix.

This expression can be derived in many ways. One of them is by minimizing the cost function

$$\mathbf{J}_k(\mathbf{X}_k) = (\mathbf{Y}_k - \mathbf{H}\mathbf{X}_k)^T \mathbf{R}_k^{-1}(\mathbf{Y}_k - \mathbf{H}\mathbf{X}_k) + (\mathbf{X}_k - \mathbf{X}_k)^T (\mathbf{P}^f_k)^{-1}(\mathbf{X}_k - \mathbf{X}_k) \tag{20}$$

with respect to $\mathbf{X}_k$.

The Kalman gain minimizes the variance of the error of the analyzed state and describes how much value is given to the measurement or forecast state. So, if the measurement is very inaccurate, the measurement will be ignored, whereas if the estimated state is inaccurate, the analyzed state will be very close to the measurement.

Note that for this recursive scheme the initial state of the system $\mathbf{X}_0$ and the initial covariance matrix $\mathbf{P}_0$ are required. Also note that the model is used as a black box, that is, the dynamics, driving forces and numerical scheme of the model are not changed, the filter only updates the input variables of the model.

6.2 Ensemble Kalman Filter

When dealing with nonlinear dynamical systems with a large scale, the applicability of the KF becomes problematic. The KF applied to a Large scale systems involves high computational costs and memory, when calculating the covariance matrices. As of non-linearity, one solution is linearization of the model, but in many cases this is non-trivial and will increase the computational cost even more. Instead an Ensemble Kalman Filter [10] may be implemented. This filter deals with these problems by using a Monte-Carlo approach, i.e. creating an ensemble of states that will describe the initial state and statistics of the system, using the probability density of the state. In the first step of the EnKF an ensemble of N states is generated from the distribution of the initial state. Each member of the ensemble is then forwarded by the model and updated with the available measurement, using the same equations as in the KF. Whenever necessary, statistical moments are calculated with sample statistics. The mean state of all analyzed ensembles will represent the analyzed state. The fact that this algorithm does not involve linearization and each ensemble is treated the same, makes the implementation simple and therefore the the EnKF is a widely used filter.

Now the EnKF will be described more thoroughly. Considering a non-linear model the model update in equation 13 can be rewritten as

$$\begin{aligned} \mathbf{X}_{k+1} &= \mathbf{f}_k(\mathbf{X}_k) + \mathbf{G}_k\mathbf{W}_k & (21)\\ \mathbf{Z}_k &= \mathbf{H}_k\mathbf{X}_k + \mathbf{v}_k & (22) \end{aligned}$$

where $\mathbf{f}_k$ represents the nonlinear model updating function, which can be dependent on time k, but this will be committed from now on.

The EnKF scheme requires an initial ensemble. Let N be the ensemble size and $\mathbf{X}_0$ be the estimate of the initial state of the system. The initial ensemble members $\mathbf{X}_{0,i}$, where $i \in 1, .., N$, are based on $\mathbf{X}_0$. Each $\mathbf{X}_{0,i}$ is a realization of the random variable that is normally distributed with mean $\mathbf{X}_0$ and covariance matrix $\mathbf{P}_0$, which represents the uncertainty in the initial state of the system.

Now the EnKF proceeds with a time updating step for each member of the ensemble

$$\mathbf{X}^f_{k+1,i} = \mathbf{f}(\mathbf{X}^a_{k,i}) + \mathbf{G}_k\mathbf{W}_{k,i}, i \in 1, .., N \quad (23)$$

Where $\mathbf{W}_{k,i}$ are noise realizations. The mean of the forecast is

$$\overline{\mathbf{X}^f_{k+1}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}^f_{k+1,i} \quad (24)$$

The perturbations of the ensemble to the mean

$$\mathbf{L}^f_{k+1} = \left[\mathbf{X}^f_{k+1,1} - \overline{\mathbf{X}^f_{k+1}}, \mathbf{X}^f_{k+1,2} - \overline{\mathbf{X}^f_{k+1}}, ..., \mathbf{X}^f_{k+1,N} - \overline{\mathbf{X}^f_{k+1}}\right] \quad (25)$$

provide an estimate to the covariance matrix of the error in the forecast state

$$\mathbf{P}^f_{k+1} = \frac{1}{N-1}\mathbf{L}^f_{k+1}(\mathbf{L}^f_{k+1})^T \quad (26)$$

Now the Kalman gain matrix is computed in the same way as equation 18, such that the measurement update for each ensemble can be written as

$$\mathbf{X}^a_{k+1,i} = \mathbf{X}^f_{k+1,i} + \mathbf{K}_{k+1}(\mathbf{Z}_{k+1,i} - \mathbf{H}_{k+1}\mathbf{X}^f_{k+1,i}), i \in 1, .., N \tag{27}$$

where $\mathbf{Z}_{k+1,i}$ are measurement realization, according to 22. So, in this scheme not only the model is subjected to noise; the measurements are also perturbed. An application without perturbed measurements is known to result in an ensemble collapse, when the ensemble spread reduces too rapidly [28]. It was shown that updating each ensemble member using independently perturbed observations prevents an ensemble collapse. Since this revelation, the EnKF has ever after been used in the manner described above.

Finally, the analyzed state and covariance can be written as

$$\overline{\mathbf{X}^a_{k+1}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{X}^a_{k+1,i} \tag{28}$$

$$\mathbf{P}^a_{k+1} = (\mathbf{I} - \mathbf{K}_{k+1}\mathbf{H}_{k+1})\mathbf{P}^f_{k+1} \tag{29}$$

In figure 5 it is shown how a ensemble based data assimilation method is performed in our case. The initial ensemble of model states is used as input to the model. The GEM/BLOOM model runs on the cluster and as each model run is independent, they will run parallel, which makes the filter very efficient. When all ensemble members are forwarded by the model, the output will be transferred to Matlab, where the system noise is added. Also the observations are loaded into Matlab and the measurement noise is added. Then follows the measurement update in Matlab and for each ensemble new input files are generated, such that they can be forwarded by the model once again.

When dealing with a non-linear measurement function of the state variables, this EnKF scheme becomes invalid. Also, when there is a need for parameter estimation, the EnKF scheme needs to be adjusted. This can be done in a straightforward way, which will be explained now.

6.2.1 Augmented state

As was just shown, the EnKF is applicable to nonlinear models, which the KF was not. The EnKF can further be adapted as to be able to deal with systems that have a nonlinear measurement function of the state variables. In this case the system reads as

$$\mathbf{X}_{k+1} = \mathbf{f}(\mathbf{X}_k) + \mathbf{G}_k\mathbf{W}_k \tag{30}$$

$$\mathbf{Z}_k = \mathbf{h}_k(\mathbf{X}_k) + \mathbf{v}_k \tag{31}$$

where $\mathbf{h}_k$ is the non-linear measurement function.

Still, the EnKF as just described can be used, when the measured variables are simply added to the state vector, as proposed in [10]. Let $\hat{\mathbf{X}}$ be this augmented state vector

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{X} \\ \mathbf{O} \end{bmatrix} \tag{32}$$

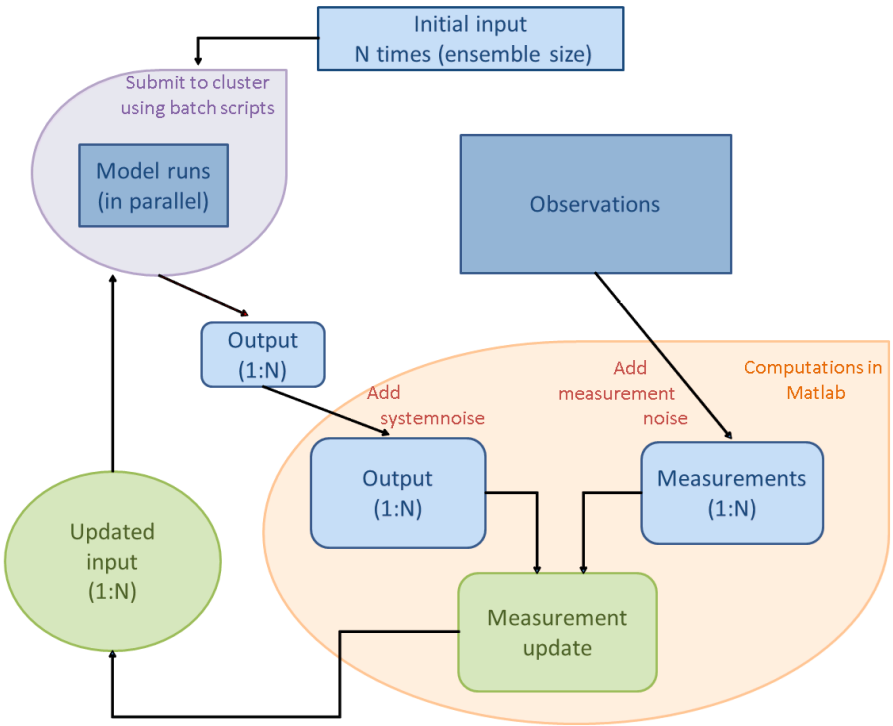


Figure 5: ***Data Assimilation scheme***

where

$$\mathbf{O} = \mathbf{h}(\mathbf{X}) \tag{33}$$

Using this augmented state makes the measurement function linear such that the system reads as follows

$$\hat{\mathbf{X}}_{k+1} = \hat{\mathbf{f}}(\hat{\mathbf{X}}_k) + \hat{\mathbf{G}}_k \hat{\mathbf{W}}_k \tag{34}$$

$$\hat{\mathbf{Z}}_k = \hat{\mathbf{H}}_k(\hat{\mathbf{X}}_k) + \hat{\mathbf{v}}_k \tag{35}$$

where $\hat{\mathbf{H}}$ is defined as

$$\hat{\mathbf{H}} = \begin{bmatrix} \mathbf{0} & \mathbf{1} \end{bmatrix} \tag{36}$$

where $\mathbf{1}$ denotes an identity matrix, such that only the output is used as a measurement, i.e.

$$\hat{\mathbf{H}}\hat{\mathbf{X}} = \mathbf{O} \tag{37}$$

Furthermore, $\hat{\mathbf{f}}$ is now defined as

$$\hat{\mathbf{f}} = \begin{bmatrix} \mathbf{f} \\ \mathbf{h} \circ \mathbf{f} \end{bmatrix} \tag{38}$$

with, $(\mathbf{h} \circ \mathbf{f})(\mathbf{X}) = \mathbf{h}(\mathbf{f}(\mathbf{X}))$, such that

$$\hat{\mathbf{f}}(\hat{\mathbf{X}}^k) = \begin{bmatrix} \mathbf{X}^{k+1} \\ \mathbf{O}^{k+1} \end{bmatrix} \tag{39}$$

6.2.2 Parameter estimation

Besides improving the prediction of the state vector, it may be of interest to also improve the parameter values. When parameters need to be estimated, it is possible to include them in the filtering procedure. They will be included in the time and measurement update. This means that the parameters need to be included in the state vector as well. Again the state vector is augmented, this time to

$$\mathbf{X}^* = \begin{bmatrix} \mathbf{X} \\ \mathbf{O} \\ \mathbf{p} \end{bmatrix} \tag{40}$$

where $\mathbf{p}$ are the parameters that need to be estimated.

Note that the model does not update the parameters. So, only a system noise for the parameters needs to be defined. Furthermore, since the parameters are not observed, the measurement function remains linear, as

$$\hat{\mathbf{H}}^* = \begin{bmatrix} \mathbf{0} & \mathbf{1} & \mathbf{0} \end{bmatrix} \tag{41}$$

such that

$$\mathbf{H}^*\mathbf{X}^* = \mathbf{O}. \tag{42}$$

In [34] the measurement update for the augmented state with parameters is derived. The augmentation approach leads to a decoupled EnKF for the state variables and the parameters. The advantage of this dual EnKF is that both states and parameters can be estimated simultaneously. Moreover, the implementation of the parameter estimation does not require changing the data assimilation algorithm; it is only supplemented by a parameter measurement update step which is similar to the state measurement update equation.

Consider the augmented state vector $\mathbf{X}^*$, the analyzed state or measurement update for the ensemble member read as

$$\mathbf{X}_i^{*a} = \mathbf{X}_i^{*f} + \mathbf{K}(\mathbf{Z}_i - \mathbf{H}^*\mathbf{X}_i^{*f}) \tag{43}$$

with

$$\mathbf{K} = \mathbf{P}^f(\mathbf{H}^*)^T(\mathbf{H}^*\mathbf{P}^f(\mathbf{H}^*)^T + \mathbf{R})^{-1} \tag{44}$$

Rewrite equation 43 in terms of the state vector and parameters

$$\mathbf{X}_i^{*a} = \begin{bmatrix} \hat{\mathbf{X}}_i^f \\ \mathbf{p}_i^f \end{bmatrix} + \mathbf{K}\left(\mathbf{Z}_i - \begin{bmatrix} \hat{\mathbf{H}} & 0 \end{bmatrix}\begin{bmatrix} \hat{\mathbf{X}}_i^f \\ \mathbf{p}_i^f \end{bmatrix}\right) \tag{45}$$

$$= \begin{bmatrix} \hat{\mathbf{X}}_i^f + \mathbf{K}^X\left(\mathbf{Z}_i - \hat{\mathbf{H}}\hat{\mathbf{X}}_i\right) \\ \mathbf{p}_i^f + \mathbf{K}^p\left(\mathbf{Z}_i - \hat{\mathbf{H}}\hat{\mathbf{X}}_i\right) \end{bmatrix} \tag{46}$$

Now the state Kalman gain $\mathbf{K}^X$ and the parameter Kalman gain $\mathbf{K}^p$ need to be

derived. Let $\mathbf{P}^f$ be the covariance matrix defined using the following expressions

$$\overline{\hat{\mathbf{X}}^f} = \frac{1}{N}\sum_{i=1}^{N}\hat{\mathbf{X}}_i^f \tag{47}$$

$$\overline{\mathbf{p}^f} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{p}_i^f \tag{48}$$

$$\mathbf{L}^X = \left[\hat{\mathbf{X}}_1^f - \overline{\hat{\mathbf{X}}^f}, \hat{\mathbf{X}}_2^f - \overline{\hat{\mathbf{X}}^f}, ..., \hat{\mathbf{X}}_N^f - \overline{\hat{\mathbf{X}}^f}\right] \tag{49}$$

$$\mathbf{L}^p = \left[\mathbf{p}_1^f - \overline{\mathbf{p}^f}, \mathbf{p}_2^f - \overline{\mathbf{p}^f}, ..., \mathbf{p}_N^f - \overline{\mathbf{p}^f}\right] \tag{50}$$

we can write

$$\mathbf{P}^f = \begin{bmatrix} \mathbf{P}_{XX}^f & (\mathbf{P}_{pX}^f)^T \\ \mathbf{P}_{pX}^f & \mathbf{P}_{XX}^f \end{bmatrix} \tag{51}$$

$$= \frac{1}{N-1}\begin{bmatrix} \mathbf{L}^X(\mathbf{L}^X)^T & \mathbf{L}^X(\mathbf{L}^p)^T \\ \mathbf{L}^p(\mathbf{L}^X)^T & \mathbf{L}^p(\mathbf{L}^p)^T \end{bmatrix} \tag{52}$$

and therefore equation 44 is equal to

$$K = \begin{bmatrix} \mathbf{P}_{XX}^f & (\mathbf{P}_{pX}^f)^T \\ \mathbf{P}_{pX}^f & \mathbf{P}_{XX}^f \end{bmatrix}\begin{bmatrix} \hat{\mathbf{H}}^T \\ 0 \end{bmatrix}\left(\begin{bmatrix} \hat{\mathbf{H}}^T & 0 \end{bmatrix}\begin{bmatrix} \mathbf{P}_{XX}^f & (\mathbf{P}_{pX}^f)^T \\ \mathbf{P}_{pX}^f & \mathbf{P}_{XX}^f \end{bmatrix}\begin{bmatrix} \hat{\mathbf{H}}^T \\ 0 \end{bmatrix} + \mathbf{R}\right)^{-1} \tag{53}$$

$$= \begin{bmatrix} \mathbf{P}_{XX}^f\mathbf{H}_X^T\left(\mathbf{H}_X\mathbf{P}_{XX}^f\mathbf{H}_X^T + \mathbf{R}\right)^{-1} \\ \mathbf{P}_{pX}^f\mathbf{H}_X^T\left(\mathbf{H}_X\mathbf{P}_{XX}^f\mathbf{H}_X^T + \mathbf{R}\right)^{-1} \end{bmatrix} \tag{54}$$

$$= \begin{bmatrix} \mathbf{K}_X \\ \mathbf{K}_p \end{bmatrix} \tag{55}$$

Note that both the state update equation and the parameter update equation have exactly the same structure as the original Kalman filter. This property makes the augmented vector based EnKF easy to implement since only one more step has to be added to the original scheme.

6.2.3 Implementation

One of the future goals is to include the parameters for *Chlfa* and K_d, see table 2, in the data assimilation. With this purpose in mind, we decided to already use the EnKF with an augmented state, as the formula for the Chlorophyll-a concentration and $K_d(490)$ are non-linear when using both parameters and state variables. This section further explains the implementation of the EnKF.

Both for the covariance matrices of the model $\mathbf{P}^f$ and measurements $\mathbf{R}$ approximations will be used in terms of the ensemble perturbations. Before, we already states to use this approximation $\mathbf{P}_e^f$ as defined in 26. Now let

$$\mathbf{\Gamma} = \{\epsilon_1, \epsilon_2, ..., \epsilon_N\} \tag{56}$$

denote the ensemble observation perturbation, or measurement noise, and approximate the measurement covariance matrix with

$$\mathbf{R}_e = \frac{1}{N-1}\mathbf{\Gamma}\mathbf{\Gamma}^T \tag{57}$$

The analysis equation, expressed in terms of the ensemble covariance matrices, now becomes

$$\begin{aligned} \mathbf{X}^a &= \mathbf{X}^f + \mathbf{P}_e^f \mathbf{H}^t \left(\mathbf{H}\mathbf{P}_e^f \mathbf{H} + \mathbf{R}\right)^{-1} (\mathbf{Z} - \mathbf{H}\mathbf{X}^f) \quad (58) \\ &= \mathbf{X}^f + \mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T \left(\mathbf{H}\mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T + \mathbf{\Gamma}\mathbf{\Gamma}^T\right)^{-1} (\mathbf{Z} - \mathbf{H}\mathbf{X}^f) \quad (59) \end{aligned}$$

Note that the inverse of

$$\mathbf{H}\mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T + \mathbf{\Gamma}\mathbf{\Gamma}^T \quad (60)$$

needs to be calculated in the measurement update.

In [10] a practical formulation and implementation is given for large systems with a large number of measurements. The inverse of matrix 60, the innovation covariance matrix, is potentially singular. Singularity of this matrix occurs when the number of ensembles is smaller then the number of measurements, which fails to make the innovation covariance matrix of full rank. Therefore, it is required to use a pseudo inverse. In order to derive a pseudo-inverse, it is assumed that

$$\mathbf{H}\mathbf{L}^f \mathbf{\Gamma} \equiv 0 \quad (61)$$

meaning that the forecast perturbations and the measurement errors are uncorrelated (equivalent to the common assumption of uncorrelated forecast and measurement errors), then the following is valid

$$\mathbf{H}\mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T + \mathbf{\Gamma}\mathbf{\Gamma}^T = (\mathbf{H}\mathbf{L}^f + \mathbf{\Gamma})(\mathbf{H}\mathbf{L}^f + \mathbf{\Gamma})^T \quad (62)$$

and the SVD decomposition of

$$\mathbf{H}\mathbf{L}^f + \mathbf{\Gamma} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (63)$$

can be used to define

$$\mathbf{H}\mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T + \mathbf{\Gamma}\mathbf{\Gamma}^T = \mathbf{U}\mathbf{\Sigma}\mathbf{\Sigma}^T \mathbf{U}^T \quad (64)$$

such that the inverse is easily calculated as

$$(\mathbf{H}\mathbf{L}^f (\mathbf{L}^f)^T \mathbf{H}^T + \mathbf{\Gamma}\mathbf{\Gamma}^T)^{-1} = \mathbf{U}(\mathbf{\Sigma}\mathbf{\Sigma}^T)^{-1} \mathbf{U}^T \quad (65)$$

Not only is the inverse now trivially calculated, this decomposition also allows for a practical implementation, since the calculation of large matrices can now be avoided, by using multiplications of this decomposition sequentially, see [10].

6.3 Ensemble Square Root Filter

The EnKF, as mentioned before, was initially designed without the use of measurement noise. However, when each ensemble is updated using the same observation, it was noted that an ensemble collapse could easily occur, as the model is systematically underestimated [2]. For this reason, measurement noise is added to prevent ensemble collapse. However, since this measurement noise is again estimated by the use of the ensemble, the method is prone to more sampling errors. Therefore, methods of the type Ensemble Square Root Filters were developed, for which the measurement updates are not as sensitive to sampling of the observational noise.

Methods of the ESRF type are derived using the same framework as the EnKF method. Recall that the EnKF is based on the Kalman filter equations

$$\begin{aligned} \mathbf{X}^a &= \mathbf{X}^f + \mathbf{K}(\mathbf{Z} - \mathbf{H}\mathbf{X}^f) && (66)\\ \mathbf{P}^a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f && (67)\\ \mathbf{K} &= \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1} && (68) \end{aligned}$$

where in the EnKF the forecast covariance matrix is approximated with the ensemble perturbation matrix

$$\begin{aligned} \mathbf{L}^f &= \left[\mathbf{X}_1^f - \overline{\mathbf{X}^f}, \mathbf{X}_2^f - \overline{\mathbf{X}^f}, ..., \mathbf{X}_N^f - \overline{\mathbf{X}^f}\right] && (69)\\ \mathbf{P}^f &= \frac{1}{N-1}\mathbf{L}^f(\mathbf{L}^f)^T && (70) \end{aligned}$$

Since all members of the ensemble are analyzed according to

$$\mathbf{X}_i^a = \mathbf{X}_i^f + \mathbf{K}(\mathbf{Z}_i - \mathbf{H}\mathbf{X}_i^f), i \in 1, .., N \quad (71)$$

and the ensemble mean gives the analyzed state, the EnKF naturally satisfies the Kalman filter equation 66,

However, the equality in Kalman equation 67 is not satisfied [28]. Because in the EnKF with perturbed observations the covariance equation 67 is satisfied in a statistical sense only, it results in a suboptimal filter behavior, which is particularly evident for small ensembles.

As an alternative to the traditional EnKF, methods belonging to the Ensemble Square Root Filters (ESRF) may be applied. These filters allow a deterministic update of the ensemble perturbations, so that the analysis error covariance matches the theoretical value given by the Kalman filter [2]. In contrast to the EnKF, in an ESRF both the ensemble mean and the ensemble perturbations are updated explicitly. The ensemble mean is updated by using the usual analysis equation 66, while the ensemble perturbations are updated via an explicitly calculated transformation, represented by the ensemble transform matrix $\mathbf{T}$:

$$\mathbf{L}^a = \mathbf{L}^f\mathbf{T}. \quad (72)$$

This transform matrix is, however, not unique as it can be multiplied with an arbitrary orthonormal matrix. The transform matrix that will be derived now, is used, as it has some appealing properties. This transform matrix is derived [11] by rewriting 67 as follows

$$\begin{aligned} \mathbf{P}^a &= (\mathbf{I} - \mathbf{K}\mathbf{H})\mathbf{P}^f && (73)\\ \mathbf{P}^a &= \mathbf{P}^f - \mathbf{K}\mathbf{H}\mathbf{P}^f && (74)\\ \mathbf{P}^a &= \mathbf{P}^f - \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^f && (75)\\ \mathbf{L}^a(\mathbf{L}^a)^T &= \mathbf{L}^f(\mathbf{L}^f)^T - \frac{1}{N-1}\mathbf{L}^f(\mathbf{L}^f)^T\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{L}^f(\mathbf{L}^f)^T && (76)\\ \mathbf{L}^a(\mathbf{L}^a)^T &= \mathbf{L}^f\left(\mathbf{I} - \frac{1}{N-1}(\mathbf{H}\mathbf{L}^f)^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{L}^f\right)(\mathbf{L}^f)^T && (77)\\ \mathbf{L}^a(\mathbf{L}^a)^T &= \mathbf{L}^f\mathbf{T}^s(\mathbf{T}^s)^T(\mathbf{L}^f)^T && (78) \end{aligned}$$

When $\mathbf{T}^s$ is defined as

$$\mathbf{T}^s = \left[\mathbf{I} - \frac{1}{N-1}(\mathbf{HL}^f)^T(\mathbf{HP}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{HL}^f\right]^{\frac{1}{2}}, \tag{79}$$

it satisfies 72. This transform matrix is usually called the symmetric solution, which has two important properties, it is mean preserving and has smallest analysis increments. In [28] the mean preserving property $\mathbf{T}^s\mathbf{1} = 0$ is proven, meaning that the mean of the ensemble analysis corresponds to the mean that was updated using 66. This means that no bias is introduced. Smallest analysis increments means that the analyzed perturbations are closest to the forecast perturbations and is proven in [2]. A nice property, as this makes the filter more numerically stable.

6.3.1 Implementation

The implementation of the ESRF that is used, is chosen to be similar to the EnKF, since they both use similar expressions. The mean analyzed state is updated using a similar implementation as in the EnKF. Furthermore, in the implementation of the EnKF a method was proposed to compute the inverse, or actually pseudo-inverse of

$$\mathbf{HP}^f\mathbf{H}^T + \mathbf{R} \tag{80}$$

The same method will be used in the implementation of the ESRF, as it is needed in the computation of 79. Subsequently, the square root in 79 is computed by the use of an eigenvalue decomposition.

6.4 Relative error

Now the data assimilation framework is almost complete. Only a description of the system noise needs to be given. Since all substances, or state variables, have a different magnitude, it is convenient to use a relative system noise instead of adding a certain magnitude of system noise. This makes the implementation easier and more general. In this case the model update reads as

$$\begin{aligned}
\mathbf{X}_{k+1,i} &= \mathbf{f}(\mathbf{X}_{k,i}) * (1 + \mathbf{G}_k\mathbf{W}_{k,i}) && (81)\\
&= \mathbf{f}(\mathbf{X}_{k,i}) + \mathbf{f}(\mathbf{X}_{k,i}) * \mathbf{G}_k\mathbf{W}_{k,i} && (82)\\
&= \mathbf{f}(\mathbf{X}_{k,i}) + \mathbf{G}_k\hat{\mathbf{W}}_{k,i} && (83)
\end{aligned}$$

where $*$ is the point wise multiplication and still $\mathbf{W}_k \sim N(0, \mathbf{Q}_k)$. Now a standard deviation of 0.1 in $\mathbf{Q}$ translates to a relative error of 10%. A description of the covariance matrix of the system noise follows in chapter 6. Here two methods of modelling the system noise are proposed and compared.

When including parameter estimation in the filter, also system noise has to be added to the parameters. Here a similar approach is taken, by defining a relative noise. This translated to instead of using the actual parameters in the filter, using their relative change *dp*. The parameters can then be calculated as

$p = p_0(1 + dp)$. The augmented state vector $\mathbf{X}^*$ now reads as

$$\mathbf{X}^* = \begin{bmatrix} \text{state variables} \\ \text{output} \\ \text{percentage change parameters} \end{bmatrix} \tag{84}$$

$$= \begin{bmatrix} \mathbf{X} \\ \mathbf{O} \\ \mathbf{dp} \end{bmatrix} \tag{85}$$

$$= \begin{bmatrix} \begin{pmatrix} 21 \text{ active substances}^{\text{all layers}} \\ 4 \text{ inactive substances}^{\text{bottom layer}} \end{pmatrix} \\ \begin{pmatrix} \text{Chlfa}^{\text{top layer}} \\ \text{Kd}^{\text{top layer}} \end{pmatrix} \\ \begin{pmatrix} 12 \text{ dp}^{\text{Chlfa}} \\ 16 \text{ dp}^{\text{Kd}} \end{pmatrix} \end{bmatrix} \tag{86}$$

These quantities are also uncertain, so noise needs to be defined as well. For the dp an $AR(1)$ process can be used, as proposed in [35]

$$\mathbf{dp}_{k+1} = \alpha \mathbf{dp}_k + \beta \omega_{\mathbf{k}} \tag{87}$$

where $\omega_k \sim N(0, \mathbf{I})$. In order to have, for instance, a variance of 0.1 for the $\mathbf{dp}_k$ for all k, the values $\alpha = 0.95$ and $\beta = 0.0312$ can be chosen.

Let $\mathbf{dp}_0$ be the initial percentage change of the parameters. Now to find the Gaussian noise associated with this noise process, we rewrite equation 87

$$\mathbf{dp}_{k+1} = \mathbf{dp}_k + (\alpha - 1)^{k+1} \mathbf{dp}_0 + \eta_k \tag{88}$$

$$= \mathbf{f}_p(\mathbf{dp}_k) + \eta_k \tag{89}$$

where $\eta_k \sim N(0, \mathbf{\Pi}_k)$ and $\mathbf{\Pi}_k = \beta^2 \sum_{n=0}^{k+1} (\alpha - 1)^{2n}$.

6.5 Validation

It is not advisable to directly apply the data assimilation to the real data, since the method still needs to be validated. In order to analyze the performance of a data assimilation procedure a twin experiment is performed. The resulting analyzed state will be evaluated by checking the Root Mean Square Error and the innovations. In the next sections these validations methods are explained.

6.5.1 Twin experiment

In a twin experiment, a synthetic truth is defined, which can be used in the filter. Therefore, a twin experiment provides a way to make sure that a filter converges to the right state, since the true state is fully known. The true state is generated by running the model starting with an initial state and a system noise. Next, measurement noise needs to be defined, which will describe the uncertainty of the true state and can be used in the filter. This can be done by evaluating the distribution of the output of an ensemble of model runs.

In a twin experiment, two procedures will be performed in order to validate the filter performance. Firstly, the model will be forwarded with N ensembles,

using the same system noise and initial condition as when generating the true state. The resulting model state represent the model without filtering and can be used as a reference for the filter, as the filter should outperform the model. Secondly, the filter will be applied on N ensembles, with the same initial condition and system noise. This filtering method results in an analyzed state. A scheme of this procedure, a twin experiment, is shown in figure 6.

Now the analyzed state can be compared to the true state. Similarly, the output of the model without filtering can be compared to the true state. When the analyzed state is closer to the true state than the state of the model without filtering, it may be confirmed that the data assimilation succeeded. It only remains to be checked whether the filter converges to the right state, i.e. if the statistics agree with the statistic given by the filter specifications. This can be done by evaluating the so-called innovations.

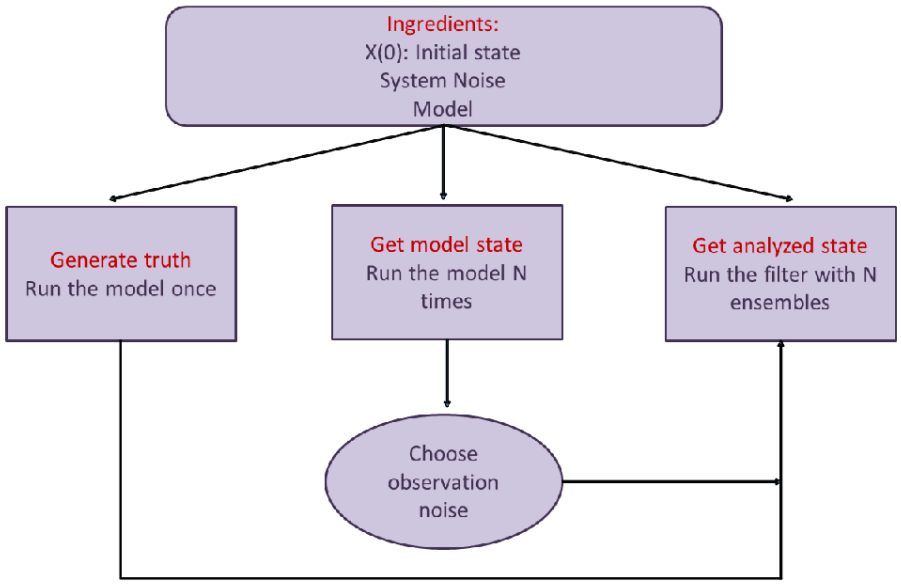


Figure 6: ***Twin experiment***

For the experiments it was chosen to only use measurements of the *Chlfa* in the filter. Furthermore, parameter estimation was not performed, so parameters were also not perturbed. This was done, so that validating the results was easier. However, it will still be investigated whether the filter improves the prediction of K_d. Extending the filter such that the K_d is also used as a measurement and adding parameter estimation is implemented easily. But to keep things simple, it has been omitted from the experiments in this thesis. Validation, however, will take place for *Chlfa* and K_d, so that we can investigate if the filter using only measurement of *Chlfa* can already improve the prediction of K_d as well. Both variables are validated on the surface of the North Sea. This measurement domain is chosen, since MERIS data is only available on the surface. To furthermore confirm with the MERIS application, the time update steps are one day, as MERIS data is available daily. The twin experiment will cover the months of the spring bloom, from mid February to the end of April.

6.5.2 RMSE

In order to evaluate how close the model or the filter is to the true state, the Root Mean Square Error is used, which is a well known cost function. Throughout this report three RMSE values are investigated. An overall RMSE, a spatial RMSE and a time series RMSE. Let X denote the predicted value and let Y denote the true state, then we have

$$RMSE = \sqrt{\frac{1}{N_t \cdot N_s} \sum_{s=1,t=1}^{N_s,N_t} (X_s^t - Y_s^t)^2} \tag{90}$$

$$RMSE_{\text{time } t} = \sqrt{\frac{1}{N_s} \sum_{s=1}^{N_s} (X_s^t - Y_s^t)^2} \tag{91}$$

$$RMSE_{\text{segment } s} = \sqrt{\frac{1}{N_t} \sum_{t=1}^{N_t} (X_s^t - Y_s^t)^2} \tag{92}$$

where N_t and N_s are the number of time points and segment, respectively. The RMSE gives the overall performance of the filter. The spatial RMSE shows how the filter performs spatially, so for each segment it can be evaluated how well the filter performs. The RMSE time series shows how well the filter performs each day, taking into account the whole model grid. For both the model and the filter, these three RMSE values are calculated. It is expected that the filter produces a lower RMSE than the model.

6.5.3 Innovations

An innovation [15] is defined as the difference between the observation and the forecast, which is defined as the mean of the ensemble of forecasts. For each time point k the innovation vector is therefore defined as

$$\mathbf{I}_k \equiv \mathbf{Y}_k - \mathbf{H}_k \overline{\mathbf{X}_k^f} \tag{93}$$

where $\mathbf{Y}_k$ is the observation vector. In [15] it is shown that

$$\mathbf{I}_k \mathbf{I}_k^T = \mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R} \tag{94}$$

which provides us with a theoretical variance for each $\mathbf{I}_k$. Also known is that $E[\mathbf{I}_k] = 0$. Now, the innovations can be evaluated against these statistics to see if the filter is consistent and filter divergence can be detected if it occurs.

The diagonal of the innovation covariance matrix

$$\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R} \tag{95}$$

is used to evaluate the performance of the filter for each segment. The innovation time series of a segment i should be Gaussian with mean zero and have variance σ_i^2 given by the diagonal entry of the innovation covariance matrix. To check if the spread is consistent, the percentage between the bounds $-\sigma_i$ and σ_i should be approximately 67% and the percentage between the bounds $-2\sigma_i$ and $2\sigma_i$ should be approximately 96%.

Filter divergence can happen in two different ways. When the calculated percentages of innovations between the theoretical bounds are lower than should be, the filter diverges from the true state. In this case, the model is overestimated. This means that the system noise prescribed to the model is too small. This type of filter divergence can be dealt with by increasing the system noise.

The calculated percentages can also be too large. When this occurs, the filter seems to converge, but the innovations tell us that it converges to the wrong state, since the statistics are inconsistent. In this case the model is underestimated and is just not taken into account enough. The filtered state does not diverge away from the true state, but collapsed into the true state. Therefore, it needs to be checked whether the spread in the ensemble forecasted states is still sufficient, since ignoring the model state and relying on the measurement too much may have lead to an ensemble collapse.

In [22] another way of checking innovation statistics is described. It is stated that the Normalized Innovation Squared (NIS) given by

$$NIS = \mathbf{I}_k(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{I}_k^T \tag{96}$$

should follow a Chi-square distribution with m, the number of measurements, degrees of freedom. However, when the filter diverges, the NIS will no longer follow a Chi-square distribution. Therefore, one can check if the NIS falls into the $1-\alpha$, with for instance $\alpha = 0.05$, confidence region for a Chi-square random variable with m degrees of freedom, to judge if the filter diverges.

However, in our case it turns out that the covariance matrix for the innovations is often singular or nearly singular, such that the equation 96 gives unrealistic values. For the filter this (near) singularity is not an issue, since the usement of the pseudo-inverse still leads to a sensible measurement update [10].

Also, the first proposed idea of checking innovations, by using the σ bounds, gives values for each segment. Therefore, this provides a way to check the performance of a filter spatially. When working with the NIS value, this is not possible. For these reasons, it is chosen to use the σ bounds to evaluate the innovations.

7 Model of the system noise

This chapter focuses on modelling the system noise on the state variables. System noise is an important aspect of Data Assimilation, as it influences the filter performance greatly. When the modelled system noise describes the real uncertainty well, the filter performs better.

A common way of defining the system noise is by choosing a covariance function that describes the covariance between segments in the model grid. What the actual covariance function is depends on the physics of the problem. However, this covariance function is generally unknown and data is often insufficiently present. A usual guess is the squared exponential covariance function, experience has shown that this function works well. Since the state vector is very large, the covariance matrix resulting from the squared exponential covariance function is very large. This property leads to some problems, which will be stated in the next section.

For this reason, and also for comparison considerations, another method of constructing the system noise is regarded as well. In this case correlated noise will be generated from uncorrelated noise by interpolation along the grid.

Resulting covariance structures from these two methods will be computed and compared. Also, the computation time and adaptability of parameters of the two methods will be taken into account. Most importantly, both models for the system noise will later on be validated by evaluating the filter performances.

7.1 Squared exponential correlation function

A way to define the system noise is by estimating its covariance matrix, $\mathbf{Q}$. The model consists of twelve layers of 4350 segments, this makes the dimension of $\mathbf{Q}$ to be 52200×52200, as it contains the covariance between each pair of segments. The matrix $\mathbf{Q}$ has a block structure of twelve layers, i.e. $\mathbf{Q} = [\mathbf{Q}_{i,j}]$ with $i, j \in \{1, 2, ..., 12\}$. Each block $\mathbf{Q}_{i,j}$ denotes the covariance between the layers i, j. We assume that the variance is isotropic, such that our covariance matrix is of the form

$$\mathbf{Q} = \sigma \mathbf{\Sigma} \sigma^T$$

where σ is a constant diagonal matrix, representing the standard deviation of the system noise and Σ is the correlation matrix.

The coefficients of the correlation matrix Σ are assumed to be defined by the squared exponential correlation function. The structure of Σ is identical to the structure of $\mathbf{Q}$; a block structure with twelve layers. So, $\mathbf{\Sigma}_{i,j}$ defines the correlation between the layers i, j. The components of such a block are given by

$$\rho_{i,j}^{s,t} = e^{-\frac{1}{2}\left(\left(\frac{X_s^i - X_t^j}{\lambda_x}\right)^2 + \left(\frac{Y_s^i - Y_t^j}{\lambda_y}\right)^2 + \left(\frac{Z_s^i - Z_t^j}{\lambda_z}\right)^2\right)}$$

where $s, t \in \{1, 2, ..., 4350\}$. In this function, for instance, X_i^s is the X coordinate of segment s is layer i and λ_x is 10% of the maximal distance between segments in the X direction. Similarly for Y and Z. The 10% is chosen from experience. In figure 8 the correlation function in the top layer is shown for four different segments.

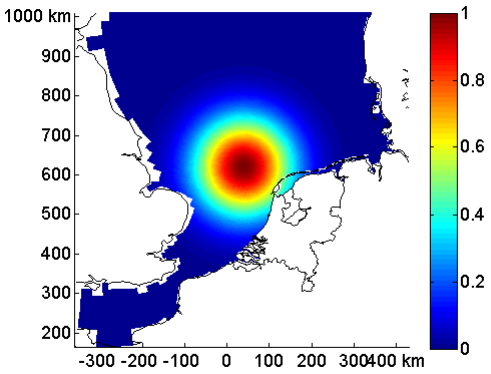


Figure 7: ***correlation with segment 1936***

Figure 8: ***Examples of correlations with a segment in the first layer. Correlation is modelled by the squared exponential correlation function.***

Now the system noise is given by $N(0, \mathbf{Q})$. To compute a corresponding noise realization, we need the square root of the correlation matrix as this enables application of the property $\sigma\sqrt{\mathbf{\Sigma}}N(0, \mathbf{I}) \sim \sigma N(0, \mathbf{\Sigma}) \sim N(0, \sigma\mathbf{\Sigma}\sigma^T) \sim N(0, \mathbf{Q})$ and save expensive computations. An eigenvalue decomposition can provide a square root of a matrix, since the correlation matrix is symmetric positive definite. The decomposition $\mathbf{\Sigma} = \mathbf{USU^T}$ gives $\sqrt{\mathbf{\Sigma}} = \mathbf{U}\sqrt{\mathbf{S}}$.

There are, however, some issues involved in the computation of the square root of the correlation matrix. First of all, in order to fit the large correlation matrix into the memory space, values below a correlation of 0.5 were set to zero. This leads to a correlation matrix which is not necessary positive definite. In this case, an eigenvalue decomposition can not be used, since eigenvalues can be negative and the resulting approximation of the correlation matrix will not be positive definite. Therefore, a singular value decomposition (SVD) will be used instead. Furthermore, since the correlation matrix is of high dimensionality, it is computational expensive to calculate all singular values. It is instead preferred to only use the first k dominant singular values to approximate the correlation matrix. This obviously leads to errors in the approximated correlation matrix. Finally, of course, numerical errors can not be avoided.

7.1.1 Approximation

To approximate the square root of the covariance matrix, the randomized power method is used [13]. In [35] this method was investigated for our application. Let Σ be the correlation matrix $n \times n$ of which the SVD is required. The randomized power method reads

- Draw an $n \times k$ Gaussian random matrix Ω.
- Form the $n \times k$ sample matrix $Y = \Sigma(\Sigma^T\Sigma)^q\Omega$.
- Form an $m \times k$ orthonormal matrix Q such that $Y = QR$.
- Form the $k \times n$ matrix $B = Q^T\Sigma$.
- Compute the SVD of the small matrix $B : B = \hat{U}SV^T$
- Let $U = Q\hat{U}$.
- Approximate Σ with USU^T.

Note that U^T is used instead of V^T to approximate the correlation matrix, this is done so that we can easily calculate the square root $\sqrt{\mathbf{\Sigma}} = \mathbf{U}\sqrt{\mathbf{S}}$.

For evaluation purposes, for now, only the first layer of the grid is considered. Using this smaller matrix $\Sigma_{1,1}$, of dimension 4350×4350, the approximation resulting from the SVD estimation will be investigated. Since the matrix consisting of twelve layers has to be made sparser in order to fit it into memory space, also a sparser version of this smaller matrix, $\Sigma_{1,1}$, will be evaluated. In this case, values below 0.25 are set to zero.

Figure 9 shows two different scenarios, as the SVD was calculated for that full matrix $\Sigma_{1,1}$, and for the sparser matrix $\Sigma_{1,1}(\Sigma_{1,1} < 0.25) = 0$. For both cases, two errors are shown. The error made by the approximation USU^T and the difference between U and V, the Euclidean norm was used. It is known that when using the approximation USV^T, the approximation becomes better when k increases. When using USU^T, also the approximation becomes better when using a larger k for the full matrix $\Sigma_{1,1}$. When applying this to the sparser matrix, the error becomes larger when increasing k, likely due to the fact that the sparser matrix is not positive definite, but it is not easy to grasp the effect of these approximations. Luckily, using a smaller k has an advantage, since computations can be made faster. However, this effect should not be disregarded, as it may become more important when the matrix is larger.

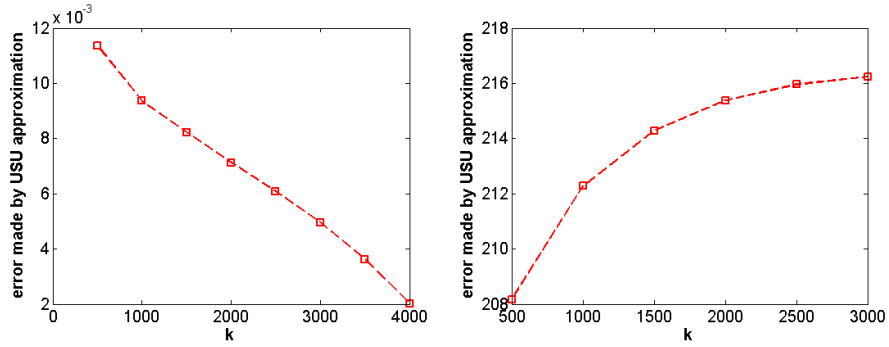
7.2 Interpolation of white noise

An interpolation of white noise can be applied to generate correlated noise. For each segment the noise will be defined by interpolating the white noise of some of the surrounding segments. This makes the noise in a certain segment dependent on the noise of its surrounding segments and therefore creates a certain correlation structure, depending on the way the contributions of the surrounding segments are defined. In this section, the general interpolation framework is described. Next, some interpolation weights are investigated and one is chosen.

7.2.1 General

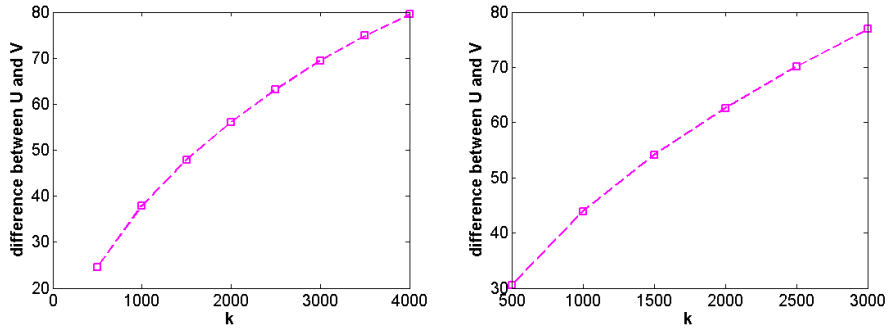
For each segment i define

$$V_i = \{j : |X_i - X_j| \leq \lambda_y, |Y_i - Y_j| \leq \lambda_y, |Z_i - Z_j| \leq \lambda_z\}$$



(a) Full matrix $\Sigma_{1,1}$

(b) Sparser matrix $\Sigma_{1,1}(\Sigma_{1,1} < 0.25) = 0$



(c) Full matrix $\Sigma_{1,1}$

(d) Sparser matrix $\Sigma_{1,1}(\Sigma_{1,1} < 0.25) = 0$

Figure 9: ***Errors made by SVD decomposition for correlation matrix first layer, using various*** k ***in the randomized power method.***

This defines the segments that are within a certain range of segment i and will thus be involved in the interpolation. The $\lambda_{x,y,z}$ are chosen equally to the correlation lengths of the squared exponential correlation function, such that the correlation scale resulting from this method will be similar. With the definition V_i the noise at segment i can now be defined by the interpolation

$$W_i = \sum_{j \in \hat{V}_i^M} \lambda_j Z_j$$

with $Z_j \sim N(0,1)$ independent.

Usually, in order to have a well defined interpolation, the restriction for the interpolation coefficients $\sum_{j \in \hat{V}_i^M} \lambda_j = 1$ is given. However, trials have shown that this results in very small correlation values. For instance, consider the variance of a segment i, which can be calculated as

$$var(W_i) = var(\sum_{j \in \hat{V}_i^M} \lambda_j Z_j) = \sum_{j \in \hat{V}_i^M} \lambda_j^2.$$

Now notice that when $\sum_{j \in \hat{V}_i^M} \lambda_j = 1$ and $\lambda_j > 0$, we have $var(W_i) \leq 1$. And this value becomes smaller when using more interpolation points.

So, another constraint for the interpolation coefficient is needed. When assuming an isotropic grid, we would like to interpolate not white noise, but unbiased noise with a constant variance, in order to obtain a desired variance after interpolation. To be able to do this, the baseline variance $var(W_i)$ needs to be equal to 1. This is easily achieved by defining the new constraint

$$\sum_{j \in \hat{V}_i^M} \lambda_j^2 = 1.$$

This is easily achieved by dividing the interpolation coefficient by

$$\sqrt{\sum_{j \in \hat{V}_i^M} \lambda_j^2}$$

This new constraint provides us the opportunity to define a fixed variance along the whole grid, by just multiplying the interpolated noise with a certain standard deviation.

7.2.2 Choosing the weights

There are several ways to choose the weights for the interpolation. Since we may assume that closer segments have more influence than segments further away, we can use inverse distance interpolation [31]. In this case the coefficients for segment i read as follows

$$\lambda_j = \frac{1}{||i-j||}.$$

Also, an exponentially decreasing weight function can be chosen

$$\lambda_j = e^{-\frac{||i-j||}{L}}.$$

Another way to define the interpolation weights is by the use of Kriging [31]. Kriging is an optimal interpolation method: the interpolation coefficients are determined in such a way that the variance in the error of interpolation is minimized. Kriging can provide another approximation to the squared exponential correlation function, instead of the SVD. In this case we again assume that the covariance between segments i, k is given by

$$\sigma^2 \cdot \rho_{i,k},$$

where ρ is the squared exponential correlation function. So, for each segment i we want the interpolation

$$W_i = \sum_{j \in \hat{V}_i^M} \lambda_j Z_j$$

to fit this covariance best. This leads to solving the following system for segment i :

$$C\Lambda = c,$$

with constraint $\sum_{j \in \hat{V}_i^M} \lambda_j^2 = 1$.

Here Λ is the vector containing the M interpolation coefficients corresponding to the M interpolation points from $\hat{V}_i^M$.

C is the $M \times M$ matrix containing the semi-variograms between the interpolation points from $\hat{V}_i^M$.

The semi-variogram is defined as

$$\gamma = \sigma^2(1 - \rho),$$

so that

$$C_{j,k} = \gamma_{j,k} = \sigma^2(1 - \rho_{j,k})$$

with $j, k \in \hat{V}_i^M$.

Finally, c is the vector containing the semi-variograms between the interpolation points from $\hat{V}_i^M$ and segment i, so $c_j = \gamma_{j,i}$,with $j \in \hat{V}_i^M$.

In order to solve this constrained minimization problem, a Lagrange multiplier is used, to deal with the constraint. This Kriging interpolation may lead to over fitting, since C is a squared matrix (same number of knowns and unknowns). Over fitting means that the interpolation fits very well to the interpolation points, but a global covariance structure will not be visible. In this case the interpolation becomes rather useless, since from a physics point of view we want a structure in our covariance. In this case a regularization term λ, needs to be included in the matrix C by using $C - \lambda I$. This gives the coefficients a bit more room and so the interpolated covariance will be more realistic.

The inverse distance weight interpolation was compared with a Kriging interpolation method. In this test, only the first layer of the grid was used. Kriging and inverse distance did not give substantially different covariances. Also the Kriging method involves a regularization parameter, which is hard to choose optimally for the larger grid of twelve layers. For this reason, simpler interpolation method weight will be chosen, either inverse distance or exponential.

In upcoming sections we will see that the interpolation method results in a seemingly similar correlation structure as the squared exponential correlation function. Therefore, we decided to investigate the resulting correlation function from the interpolation method in simple 1D examples. First consider a 1D equidistant grid model, like in figure 10.

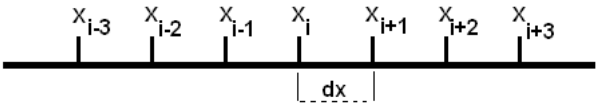


Figure 10: ***1D Equidistant grid example***

In the previous section an expression for the variance of a segment was already derived, to make sure it was equal to one. Similarly, an expression for the covariance between two segments can be derived. Let i and j be the segments of interest, $i \neq j$. Let M_i and M_j be the number of interpolation points used for segment i and j and let $\{k_1^i, ..., k_{M_i}^i\}$ and $\{k_1^j, ..., k_{M_j}^j\}$ be the corresponding interpolation points. Now the covariance between W_i and W_j can be calculated as follows

$$\begin{aligned}
Cov(W_i, W_j) &= E\left[(W_i - E(W_i))(W_j - E(W_j))\right] \\
&= E[W_i \cdot W_j] \\
&= E\left[\frac{1}{\sqrt{\sum_{s_i=1}^{M_i} \lambda_{i,k_{s_i}^i}^2}} \sum_{s_i=1}^{M_i} \lambda_{i,k_{s_i}^i} Z_{k_{s_i}^i} \cdot \frac{1}{\sqrt{\sum_{s_j=1}^{M_j} \lambda_{j,k_{s_j}^j}^2}} \sum_{s_j=1}^{M_j} \lambda_{j,k_{s_j}^j} Z_{k_{s_j}^j}\right] \\
&= \frac{1}{\sqrt{\sum_{s_i=1}^{M_i} \lambda_{i,k_{s_i}^i}^2}} \frac{1}{\sqrt{\sum_{s_j=1}^{M_j} \lambda_{j,k_{s_j}^j}^2}} E\left[\sum_{s_i=1}^{M_i} \lambda_{i,k_{s_i}^i} Z_{k_{s_i}^i} \cdot \sum_{s_j=1}^{M_j} \lambda_{j,k_{s_j}^j} Z_{k_{s_j}^j}\right] \\
&= \frac{1}{\sqrt{\sum_{s_i=1}^{M_i} \lambda_{i,k_{s_i}^i}^2}} \frac{1}{\sqrt{\sum_{s_j=1}^{M_j} \lambda_{j,k_{s_j}^j}^2}} \sum_{s:k_{s_i}^i = k_{s_j}^j} \lambda_{i,k_{s_i}^i} \cdot \lambda_{j,k_{s_j}^j} \qquad (97)
\end{aligned}$$

Where in the last equality we use that $E(Z_i Z_j) = 0$, by independence, and $E(Z_i Z_i) = 1$. So, the covariance between segment i and j is only defined by their overlapping interpolation points. Consider for instance the covariance between x_i and x_{i+1} in example figure 10 with 6 interpolation points and the inverse distance weights. The overlapping interpolation points are x_{i-1}, x_{i-2},x_{i+2} and x_{i+3}, as we cannot include x_i, since that would require a division by zero. Therefore,

$$
\begin{aligned}
Cov(W_i, W_{i+1}) &= C_i C_{i+1} \cdot \Big(\frac{1}{||x_i - x_{i-1}||}\frac{1}{||x_{i+1} - x_{i-1}||} + ... \\
&\quad \frac{1}{||x_i - x_{i-2}||}\frac{1}{||x_{i+1} - x_{i-2}||} + \frac{1}{||x_i - x_{i+2}||}\frac{1}{||x_{i+1} - x_{i+2}||} + ... \\
&\quad \frac{1}{||x_i - x_{i+3}||}\frac{1}{||x_{i+1} - x_{i+3}||}\Big) \\
&= C_i C_{i+1} \cdot \Big(\frac{1}{dx}\frac{1}{2dx} + \frac{1}{2dx}\frac{1}{3dx} + \frac{1}{2dx}\frac{1}{dx} + \frac{1}{3dx}\frac{1}{2dx}\Big) \qquad (98)
\end{aligned}
$$

where $C_i = C_{i+1} = \sqrt{\frac{1}{dx}^2 + \frac{1}{2dx}^2 + \frac{1}{dx}^2 + \frac{1}{2dx}^2}$.

Similarly,

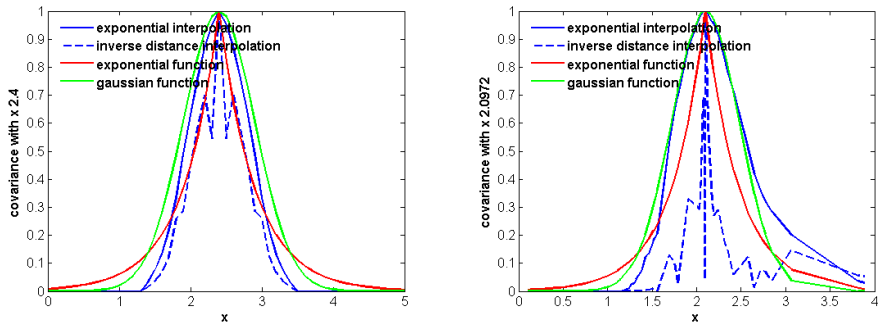
$$
\begin{aligned}
Cov(W_i, W_{i+2}) &= C_i C_{i+2} \cdot \Big(\frac{1}{||x_i - x_{i-1}||}\frac{1}{||x_{i+2} - x_{i-1}||} + ... \\
&\quad \frac{1}{||x_i - x_{i+1}||}\frac{1}{||x_{i+2} - x_{i+1}||} + \frac{1}{||x_i - x_{i+3}||}\frac{1}{||x_{i+2} - x_{i+3}||}\Big) \\
&= C_i C_{i+2} \cdot \Big(\frac{1}{dx}\frac{1}{3dx} + \frac{1}{dx}\frac{1}{dx} + \frac{1}{3dx}\frac{1}{dx}\Big) \qquad (99)
\end{aligned}
$$

Now in figure 11 we show correlation structures that resulted from a 1D grid using inverse distance weights and exponential weights. Also shown are the exponential and squared exponential correlation functions, for comparison. An example is shown for an equidistant grid and a non-equidistant grid, where the last one results in less smooth interpolated correlation structures. Note that the inverse distance interpolation leads to a correlation structure similar to the exponential correlation function. However, there are two peculiar peaks, which after investigation of 98 and 99 resulted from excluding the middle segment in the interpolation (since this would have required a division by zero). The correlation structure resulting from the exponential interpolation weights is very similar to the squared exponential correlation function.

When we investigate a 1D non-equidistant grid, the exponential interpolation weights gives a reasonable and smooth result, while the inverse distance interpolation method resulted in a non-smooth correlation structure. As we are looking for a interpolation that can approximate the squared exponential correlation function, and one that gives a smooth results, we decided to use the exponential weights for the interpolation.

Likewise to applying a covariance matrix to generate correlated noise, the interpolation method essentially smooths the white noise. For illustration purposes, in figure 12 realization of three different noises are shown. One that resulted from the SVD decomposition of the squared exponential defined covariance and one that resulted from the interpolation method using exponential weights. Also shown is a realization of white noise.

This interpolation method can be used to define the noise in all the twelve layers. The distances in z-direction are dealt with in a similar way as the SVD method, that is, one layer distance has an artificial distance of one meter. This means that segments with the same distance in X and Y have similar contributions, since an added distance of one in Z direction does not change the



(a) Correlation on equidistant grid **(b)** Correlation on non-equidistant grid

Figure 11: ***Example of correlation functions. Shown are the correlation functions resulting from interpolating white noise using inverse distance weights and exponential weights. Also shown are the exponential and squared exponential correlation (Gaussian) function. All correlation functions were calculated on an equidistant grid (a) and a non-equidistant grid (b).***

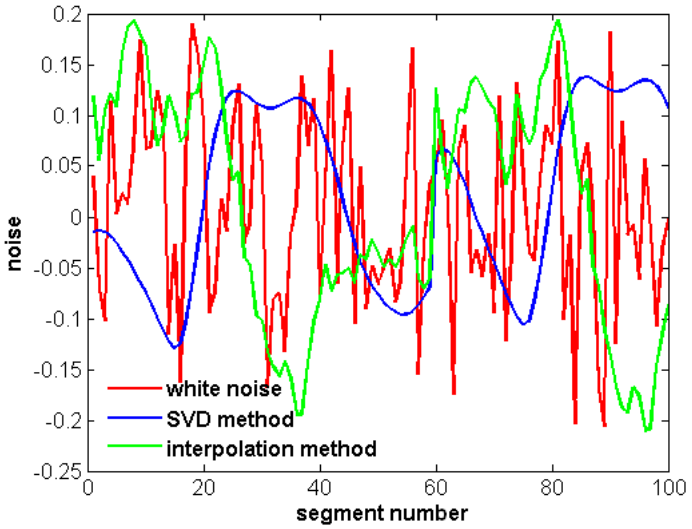


Figure 12: ***Noise realizations***

interpolation weight as much. Note that since a segments right above and under the segment of interest has a total distance of one, the contribution would be far too large. Therefore, we give these two segments an artificial distance, which is the minimal distance in all other interpolation points present.

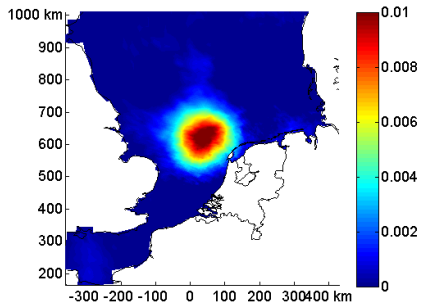
7.3 Comparison

In this section the resulting covariance structures of the system noise of the two different methods are compared. Also, the way in which both methods can be adapted to account for different correlation scales or correlation functions will be discussed. Later on, in chapter 9, both kinds of system noise are actually applied in a twin experiment, in order to find out if one of the two gives a better filter performance.

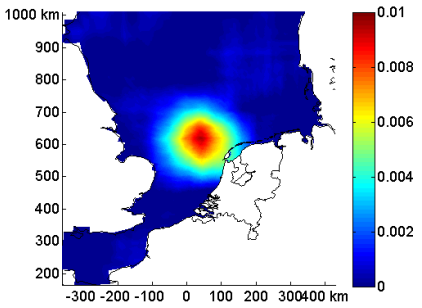
In figure 13 the covariance resulting from the two different methods are shown for segment 1936 in the first three layers. The covariance was estimated using 1000 realizations of the system noise generated by the methods. This figure shows that the covariance resulting from the SVD approximation is sometimes rather high. The maximum covariance should approximately be 0.01, but many values are 0.01 or higher. By the definition of the interpolation weights, the interpolation method does not have the same problem. However, the covariance of the SVD method seems to decrease more smoothly when the distance increases. Both methods lead to a decreased covariance, when investigating a different layer, but the covariance given by the interpolation method decreases slower per layer.

Both methods can be adapted such that longer or shorter correlation scales can be considered. They can also be adapted such that a different covariance function can be evaluated. For the SVD method, this means that the covariance matrix has to be computed again. These computations are not fast, even though we can exploit the symmetry in the covariance matrix. Subsequently, the SVD approximation needs to be computed. This goes reasonably fast, provided that we use a small k. It is, however, not yet completely clear how this k affects the approximation.

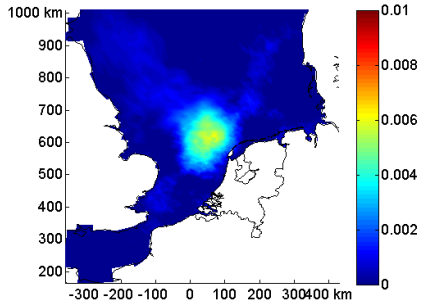
The interpolation method requires weights and a correlation scale. Calculating the weights can be an extensive computation, comparable to calculating a covariance function. The weights are currently chosen such that the resulting correlation function is smooth and is similar to the squared exponential correlation function. When other correlation function are wished to be investigated, the weights have to be adapted. This, however, can not be done in a straight forward way. It is not trivial which weights lead to which correlation structure. This has to be investigated first. Furthermore, when one would like to increase the correlation scale, the computation time increases.



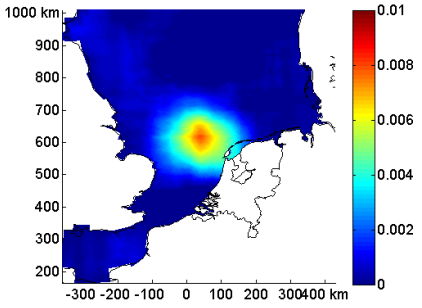
(a) SVD approximated covariance between segment 1936 and layer 1



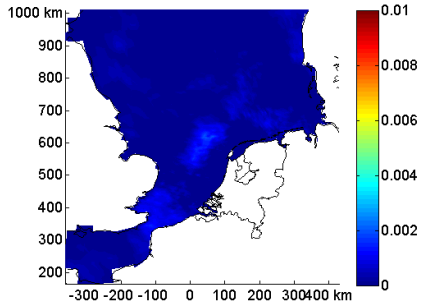
(b) Covariance from interpolation between segment 1936 and layer 1



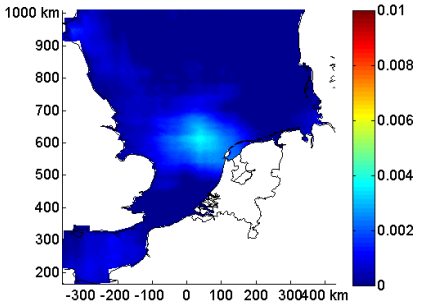
(c) SVD approximated covariance between segment 1936 and layer 2



(d) Covariance from interpolation between segment 1936 and layer 2



(e) SVD approximated covariance between segment 1936 and layer 3



(f) Covariance from interpolation between segment 1936 and layer 3

Figure 13: ***Examples of correlations with segment 1936, shown for three layers. Left shows the covariance resulting from the SVD decomposition of the squared exponential correlation function. Right shows the covariance resulting from interpolation of white noise using exponential weights. The resulting correlations were estimated by using 1000 noise realizations.***

8 Covariance Localization

The use of a small ensemble size to estimate the forecast covariance matrix, can cause long range spurious correlations. The reason for this is that the large state space is estimated by a relatively small number of ensembles. Consequently, measurements far away from a certain segment can randomly influence the measurement update of that segment. A larger ensemble size might get rid of large unwelcome correlations. However, this may prove to be impractical. Another way to avoid faulty updates is by applying covariance localization. Such methods improve the estimate of the forecast covariances, by getting rid of long range spurious correlations. There a two general ways to achieve covariance localization.

The first method multiplies occurrences of the forecast covariance matrix by a so called localization function. The other method, called localized analysis, applies the measurement update only in a certain local region around a segment. Both methods allow the removal of long range spurious correlations, and thus make sure that only measurements close by have an influence on the update of a certain segment. This chapter covers the description of both methods. For the EnKF, both methods are readily applicable. However, there still remain some impracticalities in our particular case study. As for the ESRF, one of the methods is not easily applicable. The implementation is trivial for the localized analysis method, while multiplying the covariance with a localization function, shows to be far from trivial. In the next sections all these aspects will be explained and at the end a localization method is chosen.

8.1 Localization function

This section gives a description of the method that uses a localization function to achieve covariance localization. Trials have shown some impracticalities for the EnKF, which are explained in this section. This method requires an adaption as to make it suitable for the ESRF, which is described here as well.

Recall that the measurement update is of the form

$$\begin{aligned} \mathbf{X}^a &= \mathbf{X}^f + \mathbf{K}(\mathbf{Z} - \mathbf{H}\mathbf{X}^f) && (100)\\ \mathbf{K} &= \mathbf{P}^f\mathbf{H}^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1} && (101)\\ & && (102) \end{aligned}$$

where $\mathbf{K}$ is called the Kalman gain matrix and the ensemble perturbations

$$\mathbf{L}^f = \left[\mathbf{X}_1^f - \overline{\mathbf{X}^f}, \mathbf{X}_2^f - \overline{\mathbf{X}^f}, ..., \mathbf{X}_N^f - \overline{\mathbf{X}^f}\right] \tag{103}$$

provide an estimate to the forecast covariance matrix

$$\mathbf{P}^f = \frac{1}{N-1}\mathbf{L}^f(\mathbf{L}^f)^T. \tag{104}$$

As mentioned before, this estimate may cause long range spurious correlations, which are unwelcome. Therefore, an occurrence of the forecast covariance, in the update, can be point wise multiplied with a localization function. This localization function decreases when distances between segments increase. It also

has local support, meaning that it decreases continuously to zero and outside a certain region it is constant zero. Consequently, only segments close by to each other have an influence on each other and segments farther away than a certain threshold will have no influence at all.

The point wise multiplication is also called a Schur product, denoted by a $\circ$. Using this notation, the localized Kalman gain, used in the update, is defined as

$$\mathbf{K} = \rho \circ (\mathbf{P}^f)\mathbf{H}^T(\mathbf{H}\rho \circ (\mathbf{P}^f)\mathbf{H}^T + \mathbf{R})^{-1} \tag{105}$$

It has been shown that the Schur product of two positive definite matrices is again positive definite, so the operator is valid for this application.

P^f is not actually calculated, for efficiency, so the following approximation is used to achieve localization

$$\mathbf{K} \approx \rho \circ (\mathbf{P}^f\mathbf{H}^T)(\rho \circ (\mathbf{H}\mathbf{P}^f\mathbf{H}^T) + \mathbf{R})^{-1} \tag{106}$$

In the first instance, ρ contains the localization values between the segment of the model and the observation segments. In the second occurrence, ρ contains the localization values between observation segments. Usually, [26] only one of the two occurrences of $\mathbf{P}^f$, namely $\rho \circ (\mathbf{P}^f\mathbf{H}^T)$ is used. The justification for the approximation $\rho \circ (\mathbf{P}^f\mathbf{H}^T)$ lies in that $\mathbf{H}$ is often diagonal [26].

The localization function is commonly chosen as follows

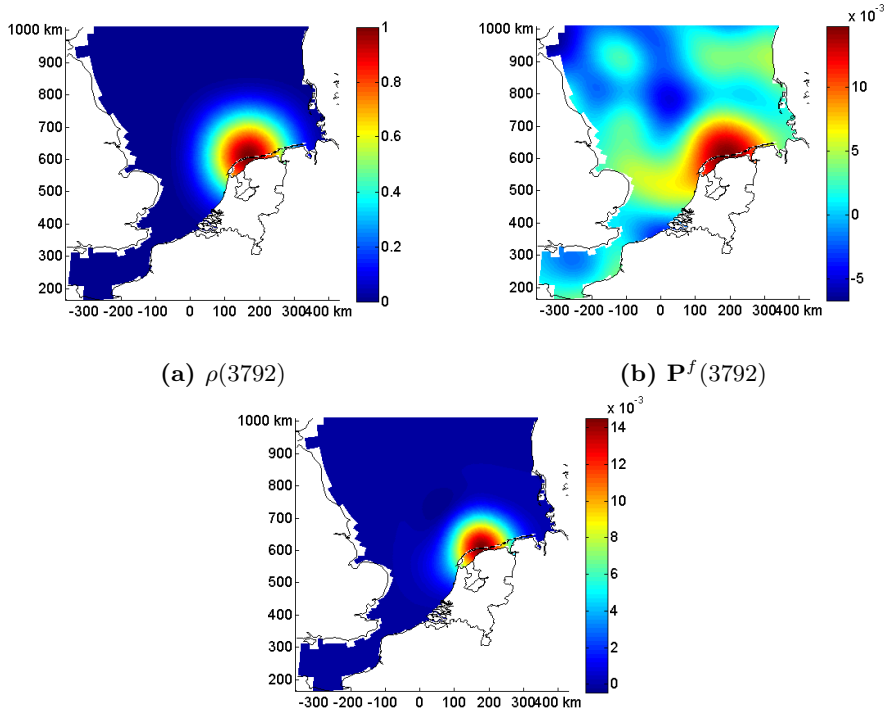
$$\rho_d = \begin{cases} -\frac{1}{4}(\frac{|d|}{c})^5 + \frac{1}{2}(\frac{|d|}{c})^4 + \frac{5}{8}(\frac{|d|}{c})^3 - \frac{5}{3}(\frac{|d|}{c})^2 + 1, & 0 \leq |d| \leq c \\ \frac{1}{12}(\frac{|d|}{c})^5 - \frac{1}{2}(\frac{|d|}{c})^4 + \frac{5}{8}(\frac{|d|}{c})^3 - \frac{5}{3}(\frac{|d|}{c})^2 - 5(\frac{|d|}{c}) + 4 - \frac{2}{3}(\frac{c}{|d|}), & c \leq |d| \leq 2c \\ 0, & 2c \leq |d| \end{cases}$$

with a chosen cut-off length l_d and usually $c_d = \sqrt{\frac{10}{3}}l_d$ is chosen. We choose to use a localization function for the X, Y and Z direction separately and multiply them: $\rho = \rho_x \cdot \rho_y \cdot \rho_z$.

An illustration of the working of this localization function is shown in figure 14. Here the localization function is shown for a certain segment. Using 30 ensembles, a covariance matrix was estimated and the small ensemble notably leads to spurious correlations. Those are removed after multiplication with the localization function.

8.1.1 Issues with the EnKF

This method can be readily implemented for the EnKF after calculation of the localization function. There is however an impractical aspect. The problem relates to calculation of the matrix $\mathbf{P}^f\mathbf{H}^T$, which is to be multiplied with the localization function. This matrix is very large, so this requires an enormous amount of extra computation time. We can use the fact that the state vector consists of several different substances, and therefore only calculate smaller parts of this matrix subsequently. Nevertheless, the computation time that is needed makes this unfeasible for the moment. It might be that parallel computations fastens this, but this is not investigated in this thesis.



(a) $\rho(3792)$

(b) $\mathbf{P}^f(3792)$

(c) $\rho(3792) \circ \mathbf{P}^f(3792)$

Figure 14: ***Illustration of the localization function working on segment 3792. Shown are the localization function (ρ) and the estimated forecast covariance ($\mathbf{P}^f$) using 30 ensembles, and the Schur product of these two.***

8.1.2 Adapted for ESRF

The ESRF updates the ensemble perturbations, that is, the ensemble square root of the forecast covariance matrix. This means that the localization function method needs to be formulated in terms of these perturbations, which turns out to be non-trivial. Two methods have been proposed by [26] and [6], both of which rely on approximations. First the general framework for the adaption is given and then follows the two proposed methods.

In order to formulate the localization function method in terms of the ensemble perturbations, the following approximation is made

$$\rho \circ \mathbf{P}^f \approx (\hat{\rho} \circ \mathbf{L}^f)(\hat{\rho} \circ \mathbf{L}^f)^T \tag{107}$$

With such a $\hat{\rho}$, the approximated localized Kalman gain in 106 can be rewritten as

$$\mathbf{K} \approx (\hat{\rho} \circ \mathbf{L}^f)(\hat{\rho} \circ \mathbf{L}^f)^T \mathbf{H}^T (\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}. \tag{108}$$

This will be used to update the analysis mean. Now the perturbation updates have to be formulated as well. In order to follow the derivation of the ESRF previously described, a further approximation is required to be

$$(\hat{\rho} \circ \mathbf{L}^f)^T \mathbf{H}^T \approx (\mathbf{H}\mathbf{L}^f)^T. \tag{109}$$

This yields the following expression for the analysis perturbations

$$\begin{aligned} \mathbf{L}^a(\mathbf{L}^a)^T &= \left(\mathbf{I} - \frac{1}{N-1}(\hat{\rho} \circ \mathbf{L}^f)(\mathbf{H}\mathbf{L}^f)^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\right)\mathbf{L}^f(\mathbf{L}^f)^T \\ &\approx (\hat{\rho} \circ \mathbf{L}^f\left[\mathbf{I} - \frac{1}{N-1}(\mathbf{H}\mathbf{L}^f)^T(\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{L}^f\right](\mathbf{L}^f)^T \end{aligned}$$

where the last step required yet another approximation, where we consider $\hat{\rho} \circ \mathbf{L}^f$ equivalent to $\mathbf{L}^f$, to allow this to be taken as a common factor. Finally, we will have

$$\mathbf{L}^a = (\hat{\rho} \circ \mathbf{L}^f)\mathbf{T}^s \tag{110}$$

where $\mathbf{T}^s$ has remained unchanged.

To apply this method, it remains to find a suitable $\hat{\rho}$. In [26] it is proposed to use

$$\hat{\rho} = \rho^{\frac{1}{2}} \tag{111}$$

It is shown that this in general does not satisfy 107. Furthermore, since $\mathbf{L}^f$ is of dimension $n \times N$, where n is the number of segments and N the number of ensembles, another approximation is needed, using only the first N vectors of $\hat{\rho}$, (equivalently, only the first N dominant singular values of ρ) .

Another way of finding a suitable $\hat{\rho}$ for 111 is given by [6]. Here an approximation is proposed by applying a Theorem (Factorization of the Schur product) that states that

If $\mathbf{P}$ is an $n \times n$ matrix such that $\mathbf{P} = \mathbf{L}\mathbf{L}^T$ with $\mathbf{L}$ of size $n \times N$ and ρ is also an $n \times n$ matrix, such that $\rho = \hat{\rho}\hat{\rho}^T$ with $\hat{\rho}$ of size $n \times M$, then

$$\rho \circ \mathbf{P} = (\hat{\rho}^T \odot \mathbf{L}^T)^T (\hat{\rho}^T \odot \mathbf{L}^T) \tag{112}$$

where $\odot$ denotes the Khatri-Rao product.

Let $\mathbf{A}$ and $\mathbf{B}$ be a matrices with r columns. Let α_i be the i-th column of $\mathbf{A}$ and β_i be the i-th column of $\mathbf{B}$. The Khatri-Rao product of $\mathbf{A}$ and $\mathbf{B}$ is the partitioned matrix

$$\mathbf{A} \odot \mathbf{B} = (\alpha_1 \otimes \beta_1 | \alpha_2 \otimes \beta_2 | ... | \alpha_r \otimes \beta_r) \tag{113}$$

where $\otimes$ indicates the Kronecker product, defined as

$$\mathbf{A} \otimes \mathbf{B} = \begin{pmatrix} a_{11}\mathbf{B} & ... & a_{1n}\mathbf{B} \\ . & & . \\ . & & . \\ . & & . \\ a_{m1}\mathbf{B} & ... & a_{mn}\mathbf{B} \end{pmatrix}$$

Note the dimensional problem, $\mathbf{L}^f$ is $n \times N$ and $\hat{\rho}^T \odot (\mathbf{L}^f)^T$ is $n \times nN$, so we need to reduce the size of $\hat{\rho}$ again, as to correctly apply the Schur-product $\hat{\rho} \circ \mathbf{L}^f$.

Since the approximation proposed by [26] is easier to compute, it is used in this thesis.

8.1.3 Issues with the ESRF

Clearly the implementation of the Schur product in the ESRF requires many approximations, in the formulation of the method and in finding a suitable $\hat{\rho}$. Furthermore, the square root of $\hat{\rho}$ is required, which is of size 52200×52200, which needs to be approximated again. All these approximations are a big disadvantage, as they might be invalid and therefore might ruin the filter.

The preferred l_x and l_y used in the localization function were 10% of the maximum distance in X and Y direction, which are the correlation distances used in creating the system noise, as we wanted to make sure that segments farther away than these correlation distances do not have an influence on the measurement update. However, in order to fit the matrix $\hat{\rho}$ into memory, the localization cut-off lengths l_x and l_y had to be chosen as one fourth of the preferred l_x and l_y. This is another disadvantage, we are limited by memory space.

8.2 Localized analysis

The other method to achieve covariance localization is called localized analysis, or domain localization [16], [10]. In this case, for every segment, the measurement update is performed separately. In such a measurement update, only measurements located near the segment of interest (inside a local region) are

used. This method only gives an approximation to the original equations that need to be solved [10]. However, it allows using a different combination of ensemble for each segment, so there is a larger flexibility to reach different states of the system.

The choice for the measurements that should be used for each segment highly depends on the application. It relies on the number of ensembles that are used and the correlation scale of the model. In [16] they recommend to make sure that the local regions of two neighboring segments greatly overlap, such that analysis does not change suddenly from one segment to another. Furthermore, they state that a reasonable approach is to use local regions of a cylinder shape and the effects of changing the width and height should be investigated. In addition, they state that local regions should correspond to the correlation distance of the model.

In the quantification of the system noise, we chose correlation lengths $\lambda_{x,y,z}$ of which are 10% of the total distance in X, Y and Z in the North Sea. As we want to make sure that segments farther away than these correlation distances do not have an influence on the measurement update, we define our local region around segment i, as follows

$$V_i = \{j : |X_i - X_j| \leq \lambda_y, |Y_i - Y_j| \leq \lambda_y, |Z_i - Z_j| \leq \lambda_z\} \tag{114}$$

Note that this leads to local regions in shapes of blocks and not cylinders. This is chosen for simplicity. The V_i are already defined, as they were used in the interpolation method of generating system noise. In figure 15 an illustration is given for this localized filter. We choose to only update the segments in the first two layers of the North Sea, as they are zero or one layer distance removed from the measurements on the surface, which is roughly 10% of all the twelve layers.

While the localization function decreases the influence of segments to zero, corresponding to their distance, the localized analysis considers all segments equally. It is however possible to decrease the influence of segments farther away in the localized analysis as well. One can let the measurement noise increase with the distance from the analysis location [16], consequently, measurements farther away have less influence. The matrix $\mathbf{R}^{-1}$ can be multiplied by a localization function, such that measurements beyond a certain distance have infinite uncertainty and will have no influence at all. This leads to a more smooth localization as opposed to solely using local regions. This was however not investigated in this thesis, because of time constraints.

8.2.1 Issues

After directly applying the localized analysis, there seemed to be a problem. Numerical instabilities occurred, leading to a blow up of the solution [12] [14]. This seemed to often occur in the Northern part of the North Sea. The reason for this is most likely that here the grid is very sparse, so local regions are small. In addition, the *Chlfa* concentrations are low, leading to a small observational noise. These two factors are named among the causes of catastrophic filter divergence [12] [14]. There does not seem to be a general threshold for which the filter consistently does not diverge. This may differ for different parameters of the ESRF, such as ensemble size and system noise. In [12] they show that

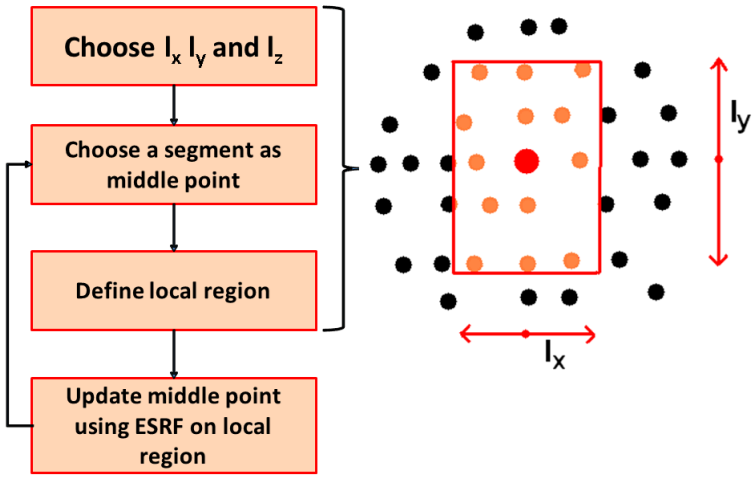


Figure 15: ***Local analysis scheme***

catastrophic filter divergence in the EnKF can be prevented by increasing the ensemble size or by applying covariance localization with the Schur product with a localization function. However, localized analysis is intended as a substitute to increasing the ensemble size and application of the Schur Product is not feasible at the moment, as explained above. Therefore, those possible solutions were not investigated. They furthermore showed that catastrophic filter divergences are suppressed by a variance limiting Kalman filter (VLKF) which controls overestimation of the analysis error covariance. This was also not investigated in this thesis, because of time constraints.

When working with MERIS remote sensing images, cloud interference results in a small number of measurements, so catastrophic filter divergence likely remains to be a problem. As a solution, we can choose to somehow constraint the measurement updates, as to avoid catastrophic divergence in general. This is investigated for the ESRF, as it turned out to outperform the EnKF in a certain important aspect (see chapter 10). Furthermore, the ESRF showed to be constrained more easily, as will be explained below.

8.2.2 Define constraints

After closer investigation of the local analysis applied to the ESRF, the divergence seems to be due to the calculation of the pseudo-inverse of

$$\mathbf{H}\mathbf{P}^f\mathbf{H}^T + \mathbf{R} \tag{115}$$

using the singular value decomposition of

$$\mathbf{H}\mathbf{L} + \mathbf{\Gamma}, \tag{116}$$

as explained in chapter 6. The matrix in equation 115 is often singular or nearly singular. The filter with a localized analysis diverges, due to small singular values of $\mathbf{H}\mathbf{L} + \mathbf{\Gamma}$. We can try to only use the singular values larger than 0.01 in the SVD, such that the SVD remains bounded.

An alternative method of constraining the update relies on a rewritten updating scheme for the ESRF [28], [32], using Woodbury's matrix identity, which is given by

$$\mathbf{K} = \mathbf{P}^a\mathbf{H}^T\mathbf{R}^{-1} \tag{117}$$

$$\mathbf{T}^s = \left[\mathbf{I} + \frac{1}{N-1}(\mathbf{H}\mathbf{L}^f)^T\mathbf{R}^{-1}\mathbf{H}\mathbf{L}^f\right]^{-\frac{1}{2}} \tag{118}$$

where

$$\mathbf{P}^a = \frac{1}{N-1}\mathbf{L}^a(\mathbf{L}^a)^T \tag{119}$$

We use the explicit form of $\mathbf{R}$ and set all values below 0.01 to 0.01, such that the inverse $\mathbf{R}^{-1}$ remains bounded.

Note that these rewritten equations for the ESRF show that catastrophic filter divergence is most likely due to small measurements. There must however be an interplay with small local regions, as when localized analysis was not applied, catastrophic filter divergence did not occur. The above defined constraints allow us to still perform the localized ESRF and test its efficiency. Both constraining methods were tested, and the one using $\mathbf{R}^{-1}$ gave good results. The other one, using the singular values, still leads to some large and unpredictable updates.

There also were some other problems related to local analysis. The eigenvalue decomposition to compute $\mathbf{T}^s$ sometimes leads to small negative values and/or small imaginary values, due to numerical errors. For now the solution was to set those value to zero. Furthermore, every segment will be updated individually, which makes the filter a bit more computational expensive. This update procedure now takes up five minutes. When larger local regions are used, this time may be increased. To speed things up, the updates for each segment can be performed in parallel, as they are independent from each other. This was not investigated in this thesis, because of time constraints.

8.3 Choice of localization method

Multiplication of the system covariance with a localization function in the EnKF turned out to be impractical due to computational time. For the domain localization on the EnKF, it is not easy to find a general way to avoid catastrophic filter divergence. Therefore, in order to choose a localization method, both localization method were performed for the ESRF. The method using a localization function turned out to be useless. The filter did not improve the non-filtered model at all. After investigation it turned out that this was due to the approximation of the square root of ρ in equation 111, as was expected. The values in this matrix square root are very small (order 10^{-3} or smaller), resulting in a measurement update where the measurements are basically ignored. Consequently, the behavior of this filter is very close to the model without filtering. Apparently, the approximations that were made are not valid.

Even though the method of domain localization, requires some extra computation time and requires some constraints, it is not as useless as the localization function method. Therefore, we decided to investigate this method more extensively and see if it improves the filter.

9 Noise propagation

In this chapter it will be investigated how the system noise on the state variables affects the output *Chlfa*. For this reason, we will evaluate the relative standard deviation of the *Chlfa*, calculated as calculated as $\frac{std}{\text{mean}(Chlfa)}$. This is expected to make more sense, since the noise on the state variables is also defined as relative. In order to evaluate the noise propagation, 30 ensembles are used. Throughout this section, the system noise defined by the SVD decomposition of the squared exponential correlation covariance matrix is applied on the state variables. The parameters are not perturbed.

We are interested in three things. First of all, how noise affects the *Chlfa* in a direct calculation, that is, without a model update. Secondly, we would like to know the same thing, but after a model update. Finally, it will be investigated how the model propagates the system noise not one day, but multiple days. This information will be used to define a measurement noise for the twin experiments.

9.1 Effect on *Chlfa*

Recall that the system noise on a state variable X is defined as a relative error

$$X^{k+1} = f(X^k) \cdot (1 + W) \tag{120}$$

where $W \sim N(0, \delta^2)$ with δ the standard deviation. For instance, $\delta = 0.1$ relates to a relative error of 10%. Now also recall that the *Chlfa* is calculated as

$$Chlfa = 1000 \cdot \sum_{i=1}^{n} ca_i \cdot ALG_i \tag{121}$$

where ca_i are positive parameters, which are assumed not to be perturbed in this case. Then

$$\begin{aligned} Chlfa &= 1000 \cdot \sum_{i=1}^{n} ca_i \cdot ALG_i \cdot (1 + W_i) && (122)\\ &= 1000 \cdot \sum_{i=1}^{n} ca_i \cdot ALG_i + 1000 \cdot \sum_{i=1}^{n} ca_i \cdot ALG_i \cdot W_i && (123)\\ &= 1000 \cdot \sum_{i=1}^{n} ca_i \cdot ALG_i + 1000 \cdot W_{Chlfa} && (124)\\ & && (125) \end{aligned}$$

where W_{Chlfa} is defined as

$$W_{Chlfa} \sim N(0, \delta^2 \cdot \sum_{i=1}^{n} ca_i^2 \cdot ALG_i^2). \tag{126}$$

such that

$$W_{Chlfa} \sim \sqrt{\sum_{i=1}^{n} ca_i^2 \cdot ALG_i^2} \cdot N(0, \delta^2). \tag{127}$$

And now because of the vector norm property $||\mathbf{x}||_2 \leq ||\mathbf{x}||_1$ we have that

$$Chlfa \leq 1000 \cdot (\sum_{i=1}^{n} ca_i \cdot ALG_i)(1 + W) \tag{128}$$

meaning that the *Chlfa* has a smaller relative error than the state variables.

In figure 16 we show the relative standard error on the *Chlfa* resulting from choosing a relative error of 10% on the state variables. The median of the relative error on *Chlfa* is 7%. This figure shows that in some regions, the standard error is quite large and in some regions very small. The spatial distribution of the standard error is very similar every day, so only one day is shown, namely March 15.

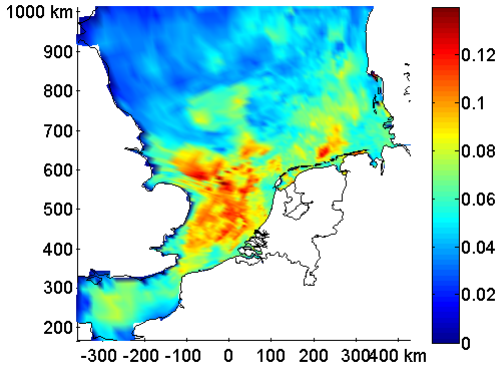


Figure 16: ***Relative standard error in Chlfa using 10% system noise, March 15.***

9.2 Effect on *Chlfa* after one day

We are also interested in the relative standard deviation of the *Chlfa* after model update of one day. Therefore, a relative error of 10% will be added to the state variables. Then all are forwarded one day by the model. Afterward, the *Chlfa* will be calculated with its relative standard deviation for each segment. The results can be found in figure 17, again only one day is shown, namely March 16. It turns out that for every day, the median value of this relative standard deviation, is around 0.03. So a relative error of 10% on the state variables translates to a relative error of approximately 0.03 on *Chlfa* after a model update. However, at some particular places the error is increased greatly.

Since again the median of the relative error on *Chlfa* is approximately the same on every day, we decided to investigate what the results are of propagating other values of the relative error on the state variables. We would like to find out if the factor of around 0.03 is constant. Therefore, we performed a model update of one day (February 14) with relative errors on the state variables of 1%, 5%, 10%, 15%, 20% and 25%. The results on the median of the relative error on *Chlfa* are found in figure 18. Here we see that indeed the relative error reduces with a factor of around 0.03.

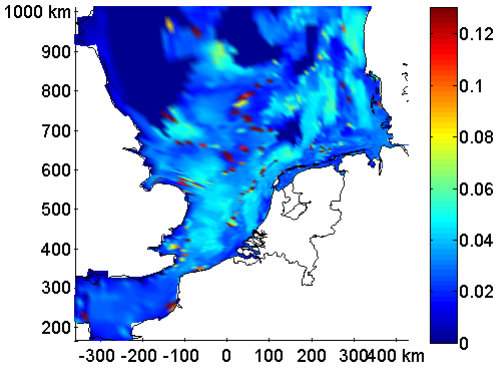


Figure 17: ***Relative standard deviation for Chlfa, March 16. Resulting from forwarding the model one day with 30 ensembles, using a 10% percentage error on the state variables.***

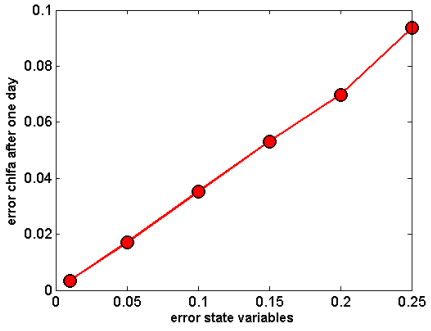


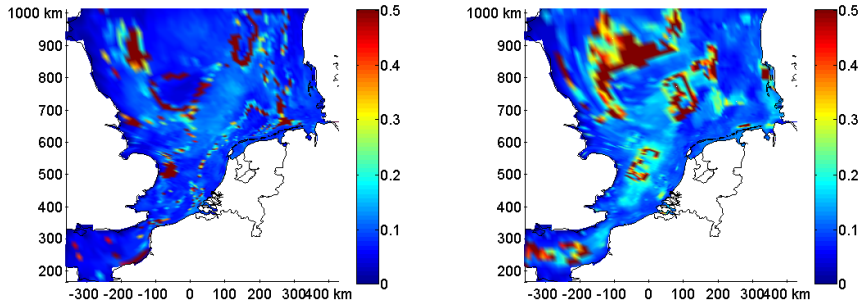
Figure 18: ***Noise propagation***

So, in conclusive, a relative error of 10% translated to a median relative error of 7% on the *Chlfa*, which after a one day model run gives a median relative error of 3% on the *Chlfa*. These ratios are constant. This information can later on be used when we need to define a system noise for the filter applied on MERIS data. For the twin experiments, however, we need to define the measurement noise. This will be done by evaluating how the noise propagation when running the model for more than one day.

9.3 Defining measurement noise

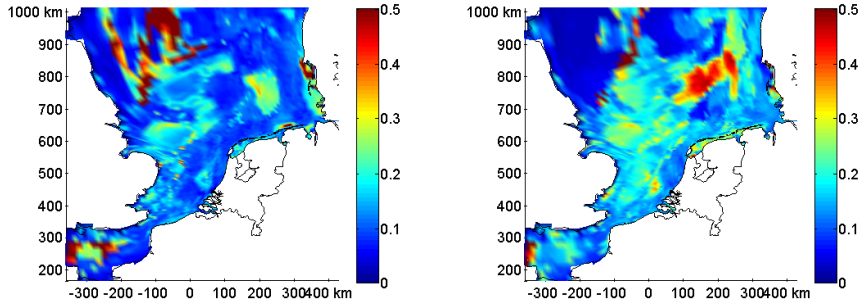
We investigated how the model propagates a system noise of 10% on the state variables, when the models runs for multiple days. After each model run of one day and adding of system noise, the *Chlfa* was calculated. And the relative error of the *Chlfa* was calculated. The results are shown in figure 19. Here we see that each day, the relative error becomes a bit larger, but is fairly constant along the grid. Note that the error becomes larger because the *Chlfa* receives an error by the propagation of the system noise, but again by adding the system noise after each day.

We would like to add a measurement noise independent on the grid, to keep things simple, so that we do not have to investigate how the measurement noise differs between segments or regions of segments. Since the relative error is fairly constant along the grid, we choose to add a relative measurement noise to the *Chlfa*. We would like the measurement noise to be constant over time, also to keep things simple. Relative standard errors for the *Chfla* were calculated for each day from February 14 to March 31. We choose a relative measurement noise of 0.1, since this is approximately the median over all days February 14 to March 31.



(a) February 14

(b) February 24



(c) March 6

(d) March 16

Figure 19: ***Relative standard deviation for the measurement noise on Chlfa, resulting from forwarding the model with 30 ensembles starting February 13, using a 10% percentage error on the state variables.***

10 Twin experiments

This section covers the results of the twin experiments that were performed in order to evaluate the performances of the filters. First the basic filter, the EnKF is evaluated. This is done in order to confirm that the EnKF improves the prediction of the model without filtering. As mentioned in chapter 2 and shown in figure 1 the purpose of the twin experiments is also to choose the best filter. RMSE values and the statistics of the innovations are considered to do so. First we will make a decision on using the EnKF or ESRF. Then the type of model for the system noise is chosen. Also, it will be investigated if domain localization improves the results and increasing the ensemble size will be investigated.

Throughout this section the same system noise, measurement noise and ensemble size are used, unless stated otherwise. The system noise is chosen as 10%, generated by the SVD method. The measurement noise is chosen as 10% as well, this was decided in chapter 8. An ensemble size of 30 is chosen, for practicality. Experience has shown that this small number is sufficient to get a good filter performance. The twin experiments all start on 13 February and end on 30 April, the months corresponding to the spring bloom. Every day a measurement update takes place, using only the *Chlfa* on the water surface.

10.1 Evaluate EnKF

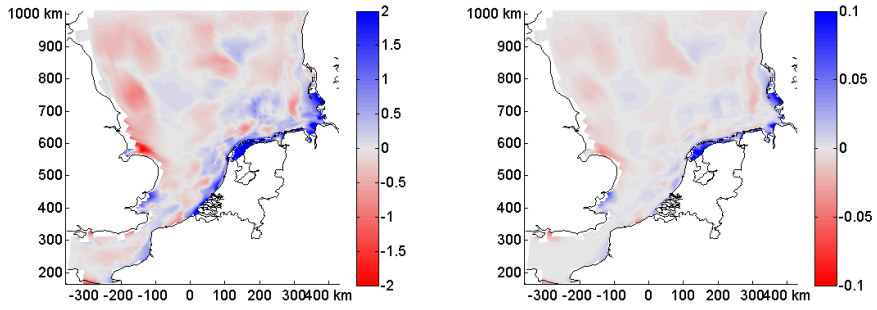
In order to evaluate whether the EnKF works well, it is compared to the model without filtering. The overall RMSE can be found in table 5. So, overall, the EnKF improves the prediction of the *Chlfa* and K_d. In figure 21 the RMSE time series and spatial RMSE are shown. Here one can clearly see that every day, the RMSE of the EnKF is significantly smaller than the RMSE of the model. When you look at the spatially distributed RMSE, you see that the EnKF does not improve the model at every segment. However, most of the largest RMSE values resulting from the model are decreased by the EnKF, especially along the coast, which is an area of great interest.

	Deterministic	EnKF
Chlfa	2.1	1.04
K_d	0.06	0.02

Table 5: ***Overall RMSE of Chlfa and Kd. Comparing the EnKF with the deterministic model.***

In figure 20 we show the quantities of the difference between the RMSE of the EnKF and model, spatially. This shows that where the EnKF performs worse than the model, the difference in RMSE is actually not that large. So, the EnKF does not worsen the prediction greatly. This figure also shows, that where the EnKF performs better than the model, it decreases the RMSE a lot.

Note that all observations that were made are valid for both *Chlfa* and K_d. So, with only using measurements of *Chlfa* the EnKF is already able to improve not only the *Chlfa*, but also the K_d.



(a) difference RMSE Chlfa **(b)** difference RMSE Kd

Figure 20: ***Spatial difference in RMSE of Chlfa and Kd. Comparing the the EnKF with the deterministic model. Blue denotes segments where the EnKF performs better than the deterministic model, red shows the opposite.***

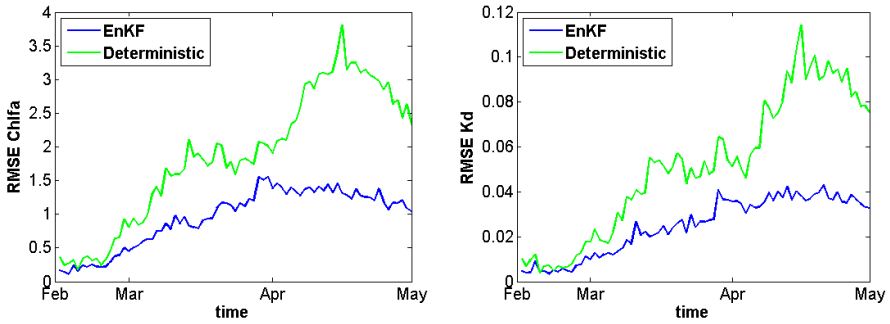
This all seems to confirm that the EnKF performs well. However, the innovations statistics need to be validated in order to see if the EnKF converges correctly. Therefore, in figure 23 we show the percentages between the σ and 2σ bounds for each segment. These percentages need to be around 67 and 96, respectively. The average percentages over the surface of the sea are 64 and 86. So, on average the statistics seem to be OK. However, the statistics are not right for every segment. Some percentages are far too low or too high. Therefore, in figure 24 the innovations of two segments are shown, for illustration purposes. The two segments are those segments where the EnKF performed the best (4327) and the worst (951). That is, where the EnKF decreased or increased the RMSE the most.

The statistics of the innovations of the segment where the EnKF performed worst are not consistent. Here the percentages (53 and 54) are too low, which indicate a filter divergence of the type where the model error is underestimated and the measurements are ignored. The *Chlfa* seems to diverge from the truth. To confirm this behavior, we validated the time series of the *Chlfa* in this segment. These are shown in figure 48 and figure 49 in the appendix. These figures show the true state of the *Chlfa*, with observations. Also, the state given by the model, with its variance is shown. Furthermore, both the states of the EnKF are shown before and after measurement update, so that we can see what the measurement update is like. Here we see that indeed the EnKF performs worse than the model. The ensemble of the EnKF forecast does not cover the truth, which explains why the measurements are mostly ignored and only the model is taken into account in the measurement update. The variance in the EnKF forecast is too small, so the model error is underestimated.

The statistics of the innovations of the segment where the EnKF performed best seem to be consistent with the theoretical bounds. The percentages (77 and 97) are close to the theoretical percentages. This tells us that the filter converges in this segment. When we look at figures 58 and figure 59 in the appendix, we see that this is indeed the case. The EnKF produces a state

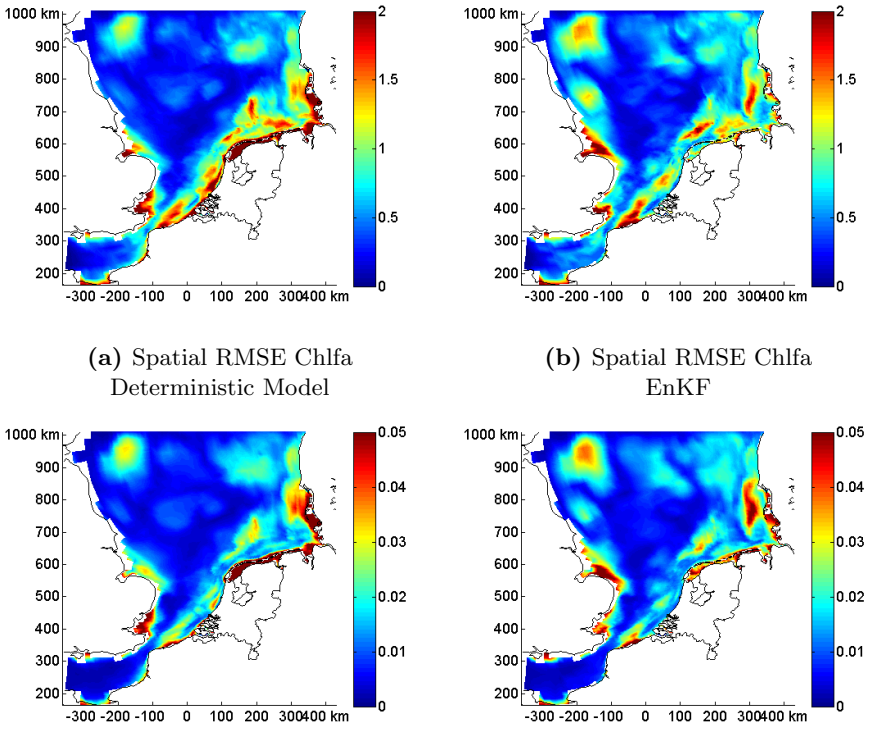
that is closer to the true state. This example segment also nicely shows that the EnKF minimizes the variance of the analyzed state, since the variance is smaller than the variance of the state before measurement update and also smaller than the observational variance.

All in all, the EnKF seems to work well. However, some things may be improved. Note, for instance, that at some segments an ensemble collapse occurred, since at some segments the percentages are too high. An example of a segment where an ensemble collapse occurred is given in figure 68 in the appendix. However, we will focus on the type of filter divergence where the EnKF diverges from the true state, which is more harmful. It may be that the more stable ESRF can improve this type of filter divergence.



(a) Timeseries RMSE Chlfa **(b)** Timeseries RMSE Kd

Figure 21: ***Timeseries RMSE of Chlfa and Kd. Comparing the EnKF with the deterministic model.***



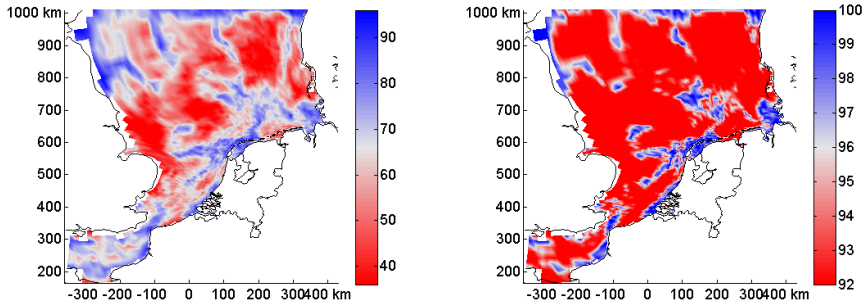
(a) Spatial RMSE Chlfa Deterministic Model

(b) Spatial RMSE Chlfa EnKF

(c) Spatial RMSE Kd Deterministic Model

(d) Spatial RMSE Kd EnKF

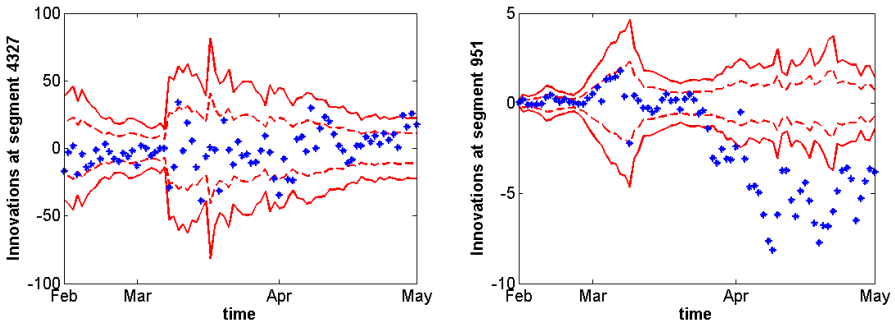
Figure 22: ***Spatial RMSE of Chlfa and Kd. Comparing the EnKF with the determinstic model.***



(a) percentage between σ bounds

(b) percentage between 2σ bounds

Figure 23: ***The percentages of the EnKF innovations between*** σ ***and*** 2σ ***bounds. Average percentages are [64,86].***



(a) Innovations segment where EnKF performs best percentages are [77,97]

(b) Innovations segment where EnKF performs worst percentages are [53,54]

Figure 24: ***Innovations of the segments were the EnKF performs best and worst. The dotted red line represents the*** σ ***bounds and the red line the*** 2σ ***bounds. The blue stars represent the estimated innovations. The first percentage, second percentage denotes the number of innovations between the*** σ ***bounds and the*** 2σ ***bounds, respectively.***

10.2 Comparison EnKF/ESRF

Now that we have confirmed that the EnKF works well, the comparison between the EnKF and ESRF can start. Both filters were performed with identical features. The ESRF is expected to perform similar, but more robust, since it is a more numerical stable filter, using small ensemble sizes. In table 6 we can see that the ESRF and EnKF indeed perform very similar. This is also visible when we evaluate the spatial RMSE, shown in figure 26. At some segments, the EnKF performs better and at other segments, the ESRF performs better. The EnKF, however, performs a bit better when we consider the time series of the RMSE, but this is likely not significant.

We also evaluate the innovation statistics of the ESRF. In figure 27 we show the percentages of innovations between σ bounds. The average of this percentage of the ESRF is 61, very similar to the value of 64 given by the EnKF. On average the percentages are similar, while spatially, the EnKF and ESRF give different percentages. In the appendix in figure 69 we show frequency histograms of the percentages of innovations between σ bounds, given by the 4350 segments on the water surface. Here we see that the ESRF is a little bit more compressed around higher percentages, indicating that it is indeed a bit more robust.

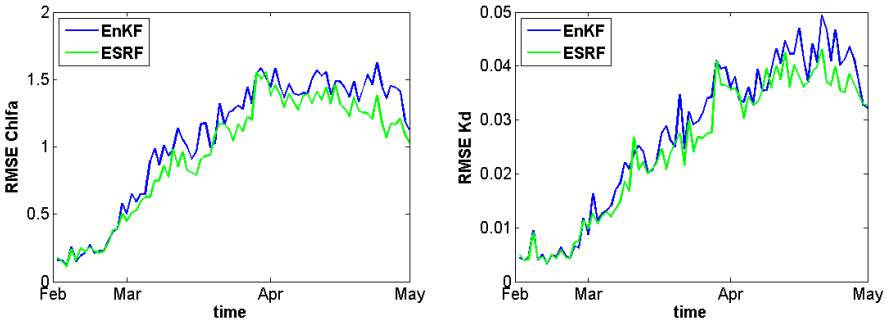
In order to get a better idea of the stability of the ESRF, we consider innovations of the same segments as in the previous section where we evaluated the EnKF. That is, the segments where the EnKF performed the best and the worst. The innovations for these segments given by the ESRF are given in figure 28. Here we see that the innovations for the segment where the EnKF performs really well, did not change that much. This is also notable in figures 60 and 61 in the appendix.

When we however consider the segment where the EnKF does not perform well, the ESRF somewhat improves the prediction. Here, still the filter does not give a very good prediction, but at least it does not diverge away from the true state. This is even better visualized in figures 50 and 51 in the appendix.

In conclusive, the ESRF and EnKF perform very similar, although a bit different spatially. The ESRF is a bit more robust to filter divergence. For this reason, we choose to use the ESRF for further investigations.

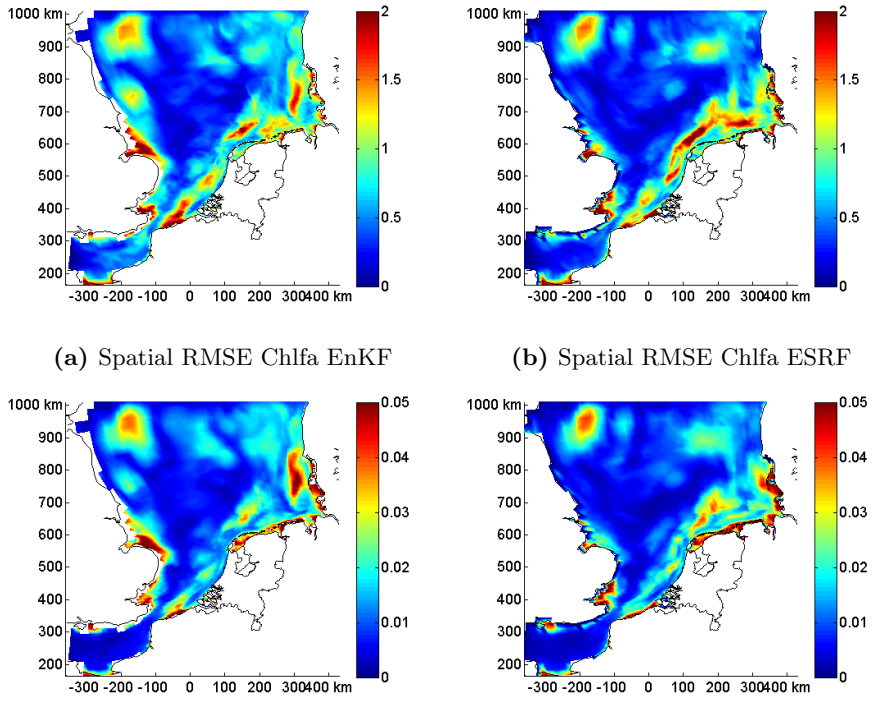
	EnKF	ESRF
Chlfa	1.04	1.16
K_d	0.02	0.03

Table 6: ***Overall RMSE of Chlfa and Kd. Comparing the ESRF with EnKF.***



(a) Timeseries RMSE Chlfa **(b)** Timeseries RMSE Kd

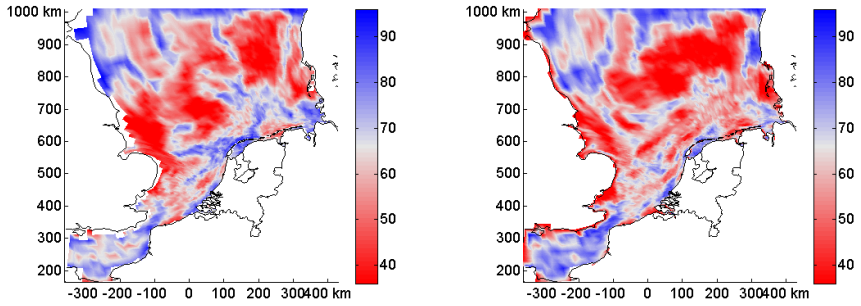
Figure 25: ***Timeseries RMSE of Chlfa and Kd. Comparing the ESRF with EnKF.***



(a) Spatial RMSE Chlfa EnKF **(b)** Spatial RMSE Chlfa ESRF

(c) Spatial RMSE Kd EnKF **(d)** Spatial RMSE Kd ESRF

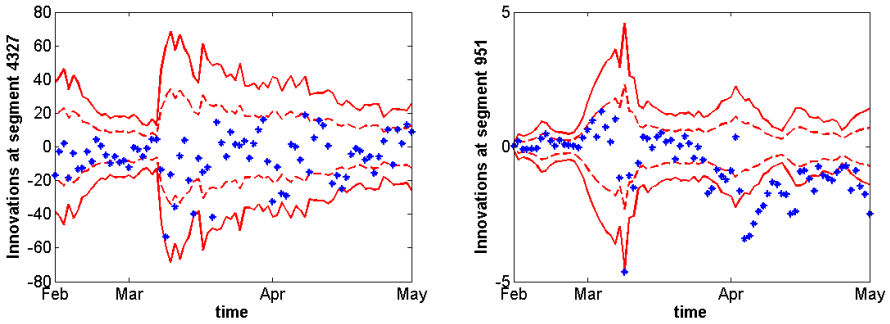
Figure 26: ***Spatial RMSE of Chlfa and Kd. Comparing the ESRF with EnKF.***



(a) percentages of innovations EnKF between σ bounds average is 64.

(b) percentages of innovations ESRF between σ bounds average is 61.

Figure 27: ***The percentages of the innovations of the EnKF and ESRF between σ bounds.***



(a) Innovations of ESRF segment where EnKF performs best percentages are [77,99]

(b) Innovations of ESRF segment where EnKF performs worst percentages are [53,67]

Figure 28: ***Innovations of the ESRF, for the segments were the EnKF performs best and worst. The dotted red line represents the σ bounds and the red line the 2σ bounds. The blue stars represent the estimated innovations. The first percentage, second percentage denotes the number of innovations between the σ bounds and the 2σ bounds, respectively.***

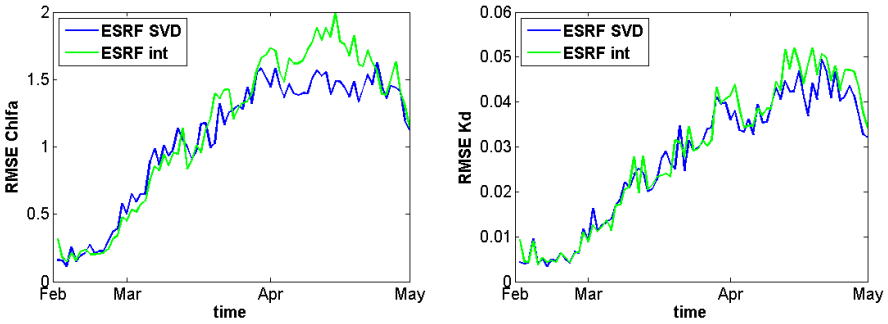
10.3 Comparison model system noise

Now the ESRF is used to compare the two models of the system noise that were described in chapter 7. The first is generated by the SVD approximation of the squared exponential correlation function (SVD) and the second one generated by interpolating white noise using exponential weights (int). Only the RMSE values are shown. The innovations do not give us more information.

Overall, as can be seen in table 7, the ESRF using system noise generated by interpolation performs similarly. The spatial distribution of the RMSE, see figure 30, is very different than the one produced by the ESRF with system noise approximated by the SVD. At most places, the ESRF with interpolated noise, does improve the RMSE, but at some places, it increases the RMSE greatly. Thus, overall, the interpolated noise does not seem to describe the uncertainty in our model much better. Therefore, the noise generated by the SVD method will be used throughout this thesis.

	ESRF SVD	ESRF int
Chlfa	1.16	1.27
K_d	0.03	0.03

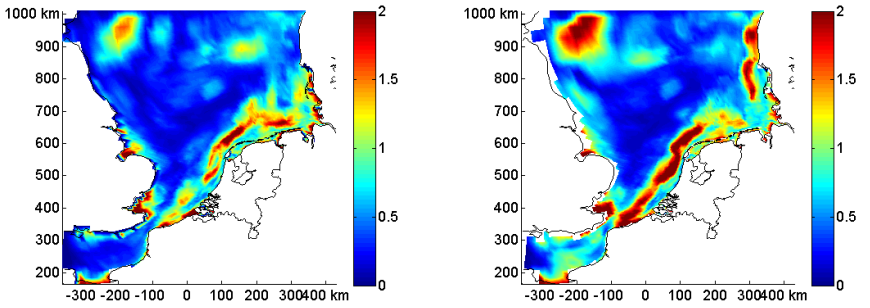
Table 7: ***Overall RMSE of Chlfa and Kd resulting from ESRF. Comparing SVD approximated squared exponential noise with interpolated white noise using exponential weights.***



(a) Timeseries RMSE Chlfa

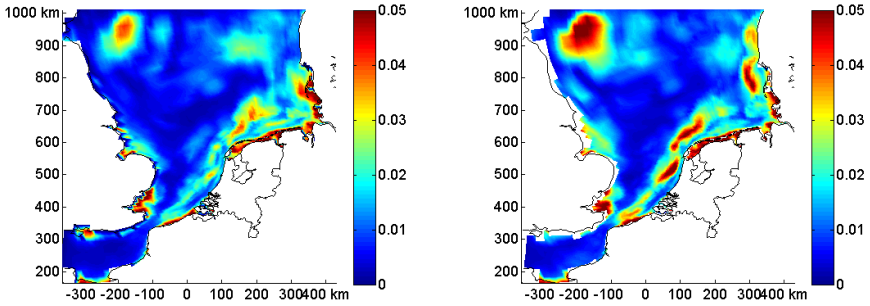
(b) Timeseries RMSE Kd

Figure 29: ***Timeseries RMSE of Chlfa and Kd resulting from ESRF. Comparing SVD approximated squared exponential noise with interpolated white noise using exponential weights.***



(a) Spatial RMSE Chlfa
ESRF
SVD approximation
squared exponential function

(b) Spatial RMSE Chlfa
ESRF
interpolated white noise
using exponential weights



(c) Spatial RMSE Kd
ESRF
SVD approximation
squared exponential function

(d) Spatial RMSE Kd
ESRF
interpolated white noise
using exponential weights

Figure 30: ***Spatial RMSE of Chlfa and Kd resulting from ESRF. Comparing SVD approximated squared exponential noise with interpolated white noise using exponential weights.***

10.4 Localization

In this part we will investigate the effect of domain localization on the ESRF, with the constraints for this method given in chapter 9. A first glance at figures 31 and 32 seem to suggest that the localization method improve the ESRF, since the RMSE is greatly reduced.

It is important to check the statistics of the innovations, to make sure that the reduced RMSE is valid. The average percentage of innovations between σ bounds is 79, which is maybe a bit high, as it should be close to 67. Overall, as seen in figure 33, the percentages are quite high. It is true that this is a sign for filter divergence, as the statistics do not correspond well with the theory. In this type of filter divergence, the filter does not trust the model enough and converges greatly to the measurements, such that an ensemble collapse occurs. This is however is less harmful than when the filter actually diverges from the true state.

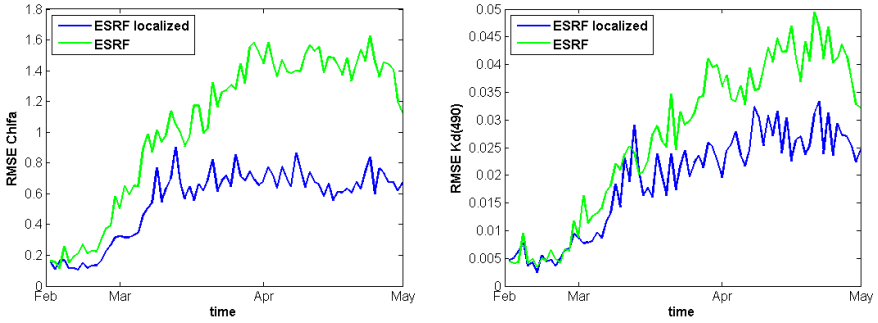
In the appendix, figures 62, 63, 52 and 53 the solution and spread of the ensemble of *Chlfa* is shown for the segments 4327 and 951. Here we see that the spread in the analyzed ensemble is quite small. Both this small variance and the high percentages of innovations between bound indicate an ensemble collapse.

As this type of filter divergence is less harmful than the other, we do not want to discard this localization method just yet. It has great potential to improve the ESRF. This method just has to be tuned more thoroughly. So, at this point, for the ensemble size investigation, we will use the ESRF, without localization.

Luckily, there was some time left to investigate how the change in some parameters of the localization changed the results. Two things were investigated to see if this can prevent the often collapse of ensembles. First, we decreased the system noise. The second thing we tried was to increase the local regions in X and Y direction with 1.5, but we still use only the first two layers in Z direction.

	ESRF	ESRF localized
Chlfa	1.16	0.6
K_d	0.03	0.02

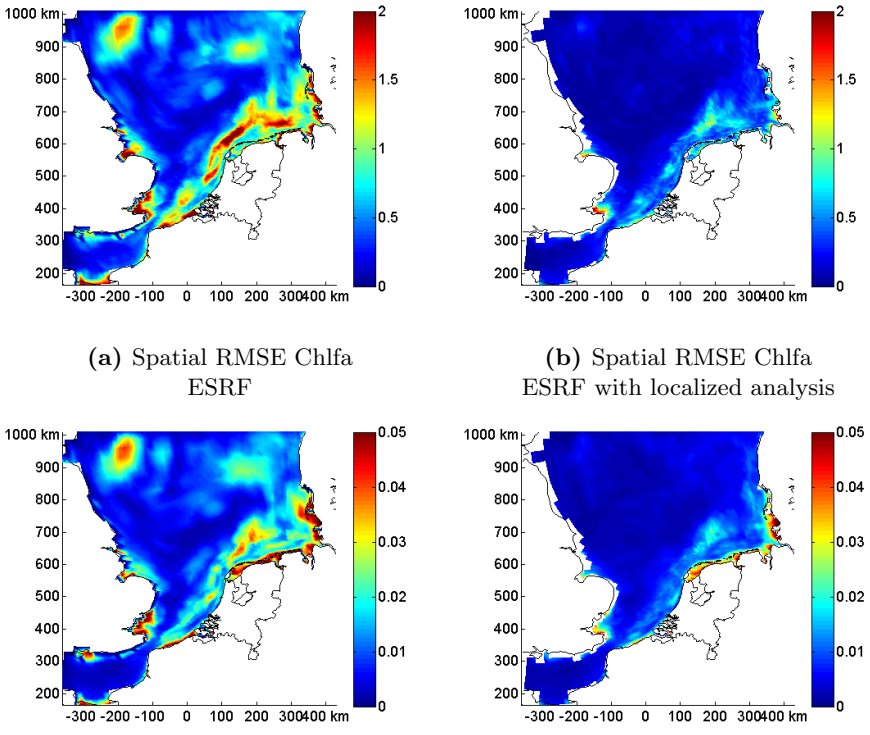
Table 8: ***Overall RMSE of Chlfa and Kd. Comparing the ESRF with the localized ESRF.***



(a) Timeseries RMSE Chlfa

(b) Timeseries RMSE Kd

Figure 31: ***Timeseries RMSE of Chlfa and Kd. Comparing the ESRF with the ESRF with localized analysis.***



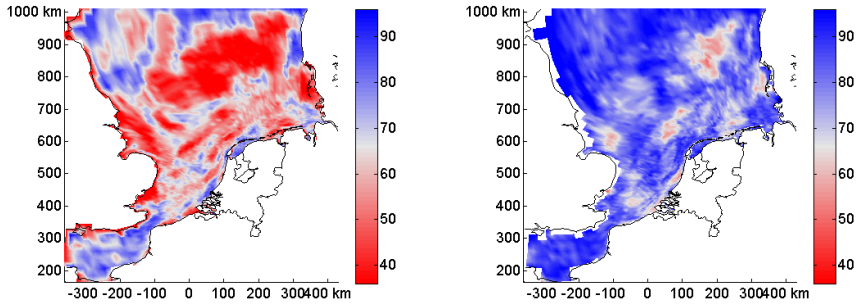
(a) Spatial RMSE Chlfa ESRF

(b) Spatial RMSE Chlfa ESRF with localized analysis

(c) Spatial RMSE Kd ESRF

(d) Spatial RMSE Kd ESRF with localized analysis

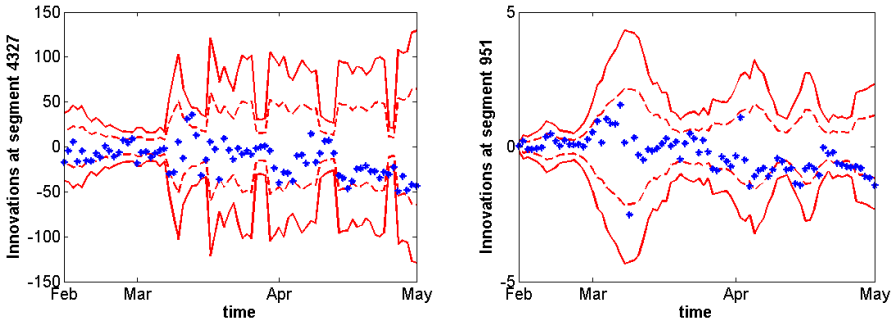
Figure 32: ***Spatial RMSE of Chlfa and Kd. Comparing the ESRF with the ESRF with localized analysis.***



(a) percentages of innovations ESRF between σ bounds average is 61.

(b) percentages of innovations ESRF localized analysis between σ bounds average is 78.

Figure 33: ***The percentages of the innovations of the ESRF and ESRF with localized analysis between σ bounds.***



(a) Innovations of localized ESRF segment where EnKF performs best percentages are [85,100]

(b) Innovations of localized ESRF segment where EnKF performs worst percentages are [91,100]

Figure 34: ***Innovations of the localized ESRF, for the segments were the EnKF performs best and worst. The dotted red line represents the σ bounds and the red line the 2σ bounds. The blue stars represent the estimated innovations. The first percentage, second percentage denotes the number of innovations between the σ bounds and the 2σ bounds, respectively.***

In table 9 we show the parameters for the three localization methods that were investigated. The first is the one for which the results were described above. The second and third both hopefully make sure that there is no ensemble collapse, since the measurements are made to be more uncertain. The second method does this by making the model more certain. The third accomplishes that measurements that largely influenced the update are less taken into account, as the local regions are increased and therefore other measurements are taken into account as well.

name method	system noise	local region
ESRF localized 1	0.1	l_x and l_y
ESRF localized 2	0.075	l_x and l_y
ESRF localized 3	0.1	$1.5 \cdot l_x$ and $1.5 \cdot l_y$

Table 9: ***Three localization methods***

In table 10 the overall RMSE values are shown and in figure 35 the spatial and time series values of the RMSE are shown. These results show that the three localized methods perform very similar, when considering the RMSE values. The localized ESRF 2 slightly outperforms the localized ESRF 3, but maybe not significantly.

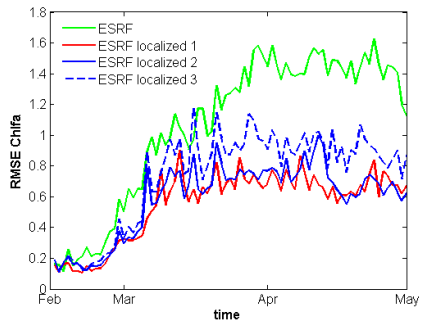
Again, we evaluate the statistics of the innovations, since the localized ESRF 1 gave a lot of segments with wrong statistics which pointed to ensemble collapse. When considering figures 36, 37 and 38 we see that both the localized ESRF method 2 and 3 give very satisfactory results in terms of the innovation statistics. This is also visible in figure 69 in the appendix, where we show a frequency histogram of the percentages of innovations between σ bounds.

In the appendix, figures 64, 65, 54 and 55 the solution and spread of the ensemble of *Chlfa* is shown for the segments 4327 and 951 given by the localized method 2. Similarly for the localized method 3 in figures 66, 67,56 and 57. Here we again see that these localized methods give similar results. The variance of the model is smaller before measurement update, such that now, the measurements are not taken into account as much and we can avoid ensemble collapse. Even though after measurement update, the variance can be very small, the innovation statistics tell us that the localized filters 2 and 3 did not diverge.

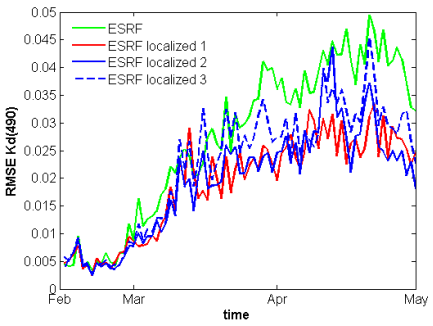
In conclusive, the performance of the localized ESRF can be greatly improved by changing some parameters. Both increasing the local regions and decreasing the system noise give good results. The RMSE values are reduced and the statistics of the innovations correspond better to the theory.

	ESRF	ESRF localized 1	ESRF localized 2	ESRF localized 3
Chlfa	1.16	0.6	0.63	0.79
K_d	0.03	0.02	0.02	0.03

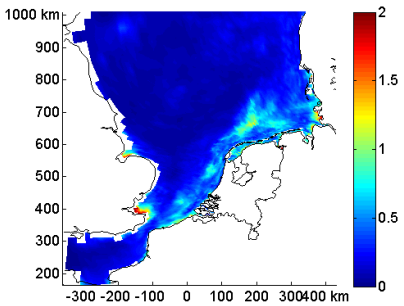
Table 10: ***Overall RMSE of Chlfa and Kd. Comparing the ESRF with the three localized ESRF methods.***



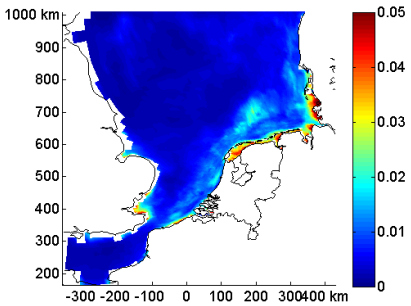
(a) Timeseries RMSE Chlfa



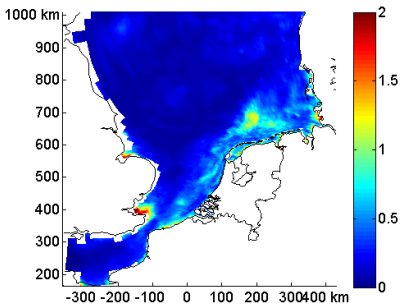
(b) Timeseries RMSE Kd



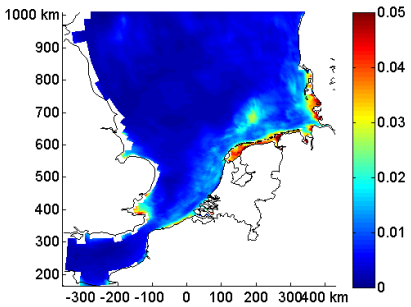
(c) Spatial RMSE Chlfa
ESRF localized 1



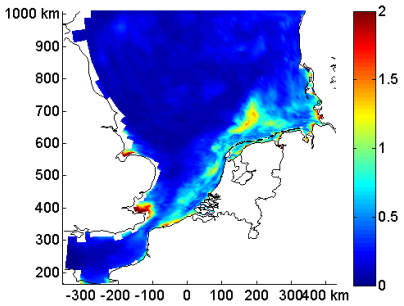
(d) Spatial RMSE Kd
ESRF localized 1



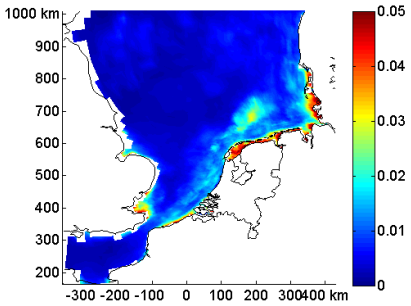
(e) Spatial RMSE Chlfa
ESRF localized 2



(f) Spatial RMSE Kd
ESRF localized 2

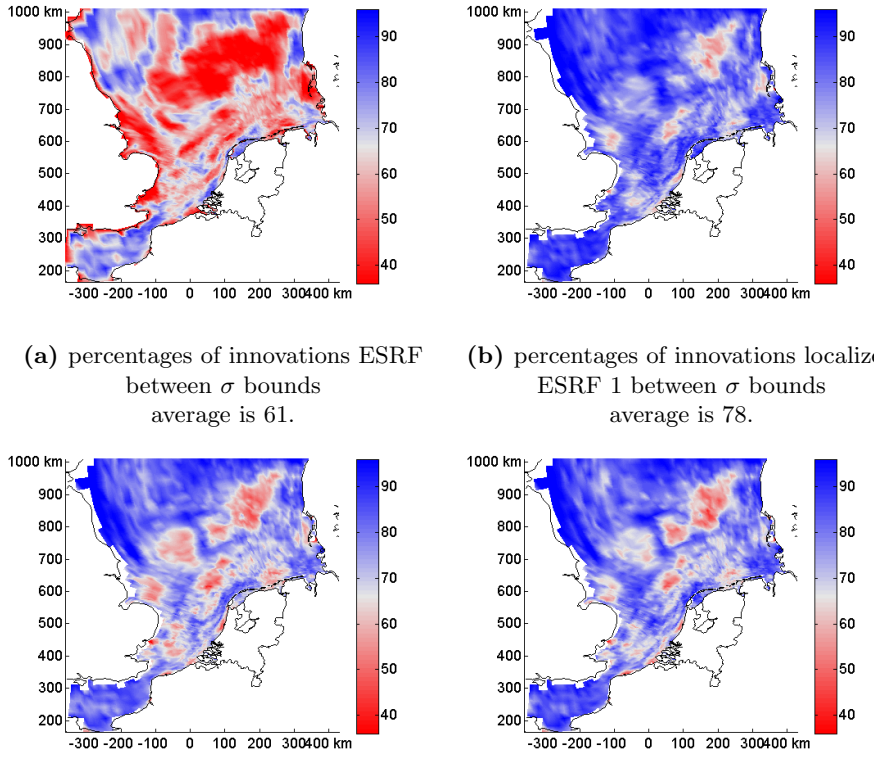


(g) Spatial RMSE Chlfa
ESRF localized 3



(h) Spatial RMSE Kd
ESRF localized 3

Figure 35: ***The percentages of the innovations of the ESRF and the three localized ESRF methods between σ bounds.***



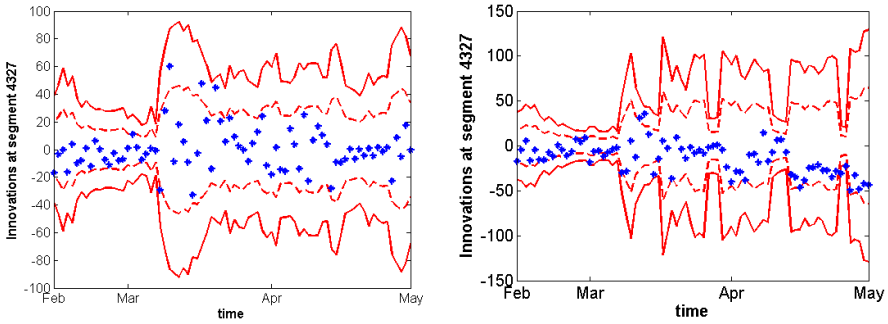
(a) percentages of innovations ESRF between σ bounds
average is 61.

(b) percentages of innovations localized ESRF 1 between σ bounds
average is 78.

(c) percentages of innovations localized ESRF 1 between σ bounds
average is 73.

(d) percentages of innovations localized ESRF 1 between σ bounds
average is 76.

Figure 36: ***The percentages of the innovations of the ESRF and the three localized ESRF methods between σ bounds.***



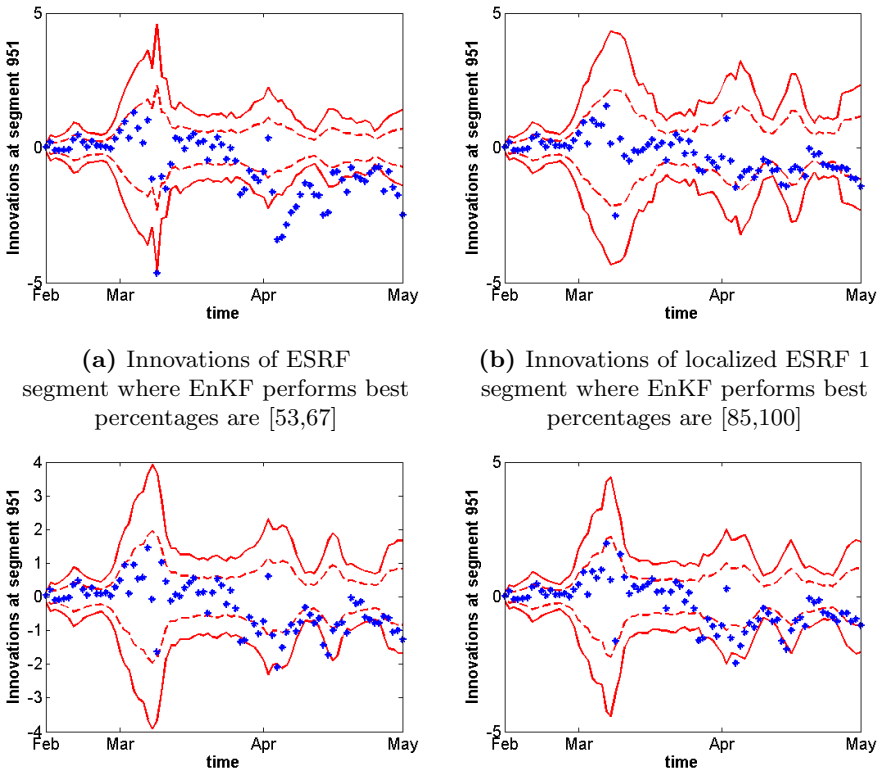
(a) Innovations of ESRF segment where EnKF performs best percentages are [77,99]

(b) Innovations of localized ESRF 1 segment where EnKF performs best percentages are [91,100]

(c) Innovations of localized ESRF 2 segment where EnKF performs best percentages are [64,90]

(d) Innovations of localized ESRF 3 segment where EnKF performs best percentages are [63,94]

Figure 37: ***Innovations of the localized ESRF, for the segment were the EnKF performs best. The dotted red line represents the σ bounds and the red line the 2σ bounds. The blue stars represent the estimated innovations. The first percentage, second percentage denotes the number of innovations between the σ bounds and the 2σ bounds, respectively.***



(a) Innovations of ESRF segment where EnKF performs best percentages are [53,67]

(b) Innovations of localized ESRF 1 segment where EnKF performs best percentages are [85,100]

(c) Innovations of localized ESRF 2 segment where EnKF performs best percentages are [65,92]

(d) Innovations of localized ESRF 3 segment where EnKF performs best percentages are [60,94]

Figure 38: ***Innovations of the localized ESRF, for the segment were the EnKF performs worst. The dotted red line represents the σ bounds and the red line the 2σ bounds. The blue stars represent the estimated innovations. The first percentage, second percentage denotes the number of innovations between the σ bounds and the 2σ bounds, respectively.***

10.5 Ensemble size

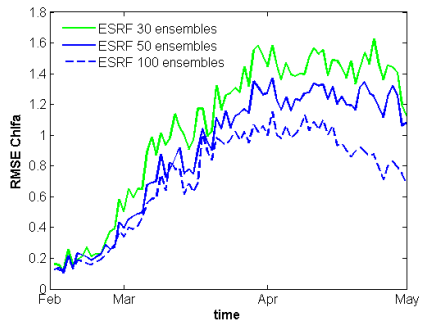
With the ESRF as our filter, the effect of increasing the ensemble size will be investigated. No localization is applied. A small ensemble size is practical, but it may be that the filter performs much better with a larger ensemble size. A trade off between practicality and better performance needs to be found. In the previous experiments, an ensemble size of 30 was applied. Now, also the results of using 50 ensembles and 100 ensembles will be investigated.

Only the RMSE resulting from the filters are shown, since the innovations statistics are very similar and do not provide us with more information. In table 11 we see that the overall RMSE does not significantly decrease when using a larger ensemble size of 50. Using 100 ensembles does improve the result more substantially. In figure 39 the time series of the RMSE are shown, as well as the spatial distribution of the RMSE. The ESRF gives a consistent lower RMSE value for the *Chlfa*, both in time and spatially, when using a larger ensemble size. The same can be said for the RMSE of the K_d, here the effects are less strong, which is not unexpected, as only observations of the *Chlfa* are used.

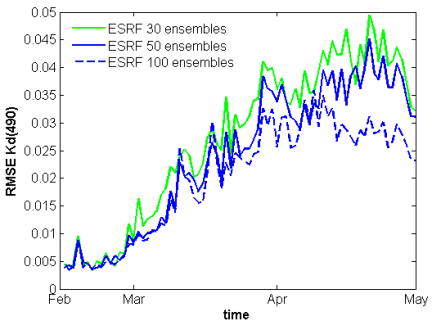
From experience it is known that using 30 ensembles still gives good results. These results seem to agree with this idea, since 50 does not improve the results substantially. Using 100 ensembles is inconveniently large and so for practical reasons, 30 ensembles will be used in our application to the MERIS data. It may be advisable to increase the ensemble size to 50 when troubles arise with 30 ensembles. However, likewise to increasing the ensemble size, localization also improves the filter. Localization is more efficient, so, first localization will be applied, before increasing the ensemble size.

ESRF	30	50	100
Chlfa	1.16	0.99	0.79
K_d	0.03	0.03	0.02

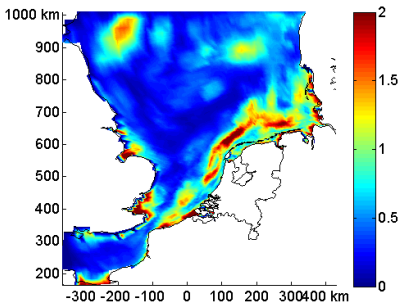
Table 11: ***Overall RMSE of Chlfa and Kd resulting from ESRF using different ensemble sizes.***



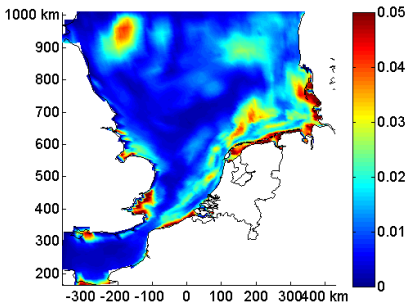
(a) Timeseries RMSE Chlfa



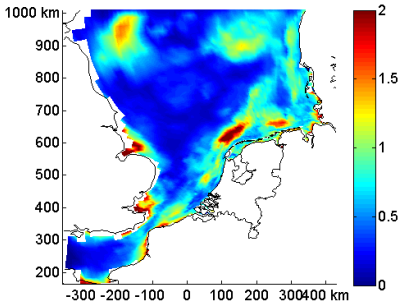
(b) Timeseries RMSE Kd



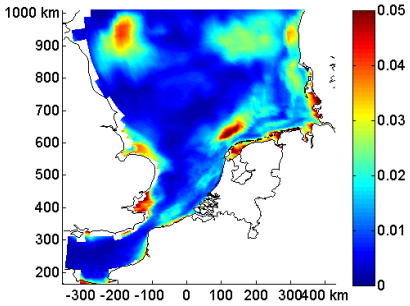
(c) Spatial RMSE Chlfa
ESRF 30 ensembles



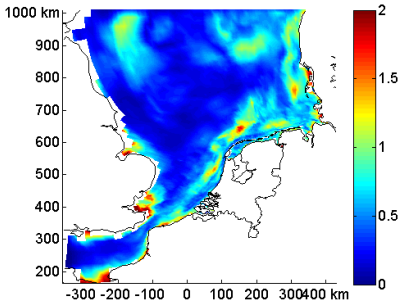
(d) Spatial RMSE Kd
ESRF 30 ensembles



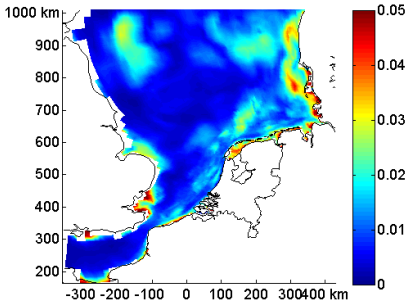
(e) Spatial RMSE Chlfa
ESRF 50 ensembles



(f) Spatial RMSE Kd
ESRF 50 ensembles



(g) Spatial RMSE Chlfa
ESRF 100 ensembles



(h) Spatial RMSE Kd
ESRF 100 ensembles

Figure 39: ***Spatial and time series of RMSE of Chlfa and Kd resulting from ESRF using different ensemble sizes.***

11 Application to MERIS data

Now that a filter is chosen with a certain system noise description and ensemble size, the investigation on applying the filter on the real data of MERIS can start. The MERIS data of the *Chlfa* comes with a standard deviation of the observation error, this will be used in the filter. As stated before, in this case, the performance of the filter will be tested by validating against the available *in situ* measurements. Only the relative error of the model still needs to be defined. This will be done by comparing the model without system noise with the MERIS data.

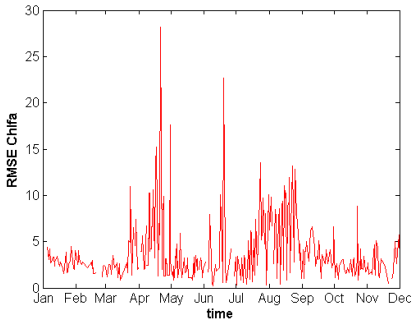
11.1 Compare Model with MERIS

To get an idea of the difference between the model (without system noise) and the MERIS data, in figure 40 we show the spatial RMSE and time series of the RMSE, when comparing the model with the MERIS data. These values are shown for *Chlfa* and for K_d. Here one can see that for the *Chlfa*, the largest differences can be found in the month March, April and May, which are actually the months belonging to the spring bloom. For the K_d a different behavior is visible, here the difference between model and MERIS seem more constant over time. When evaluating the difference between the model and MERIS spatially, one can see that the largest differences can be found along the coast, an area of great interest.

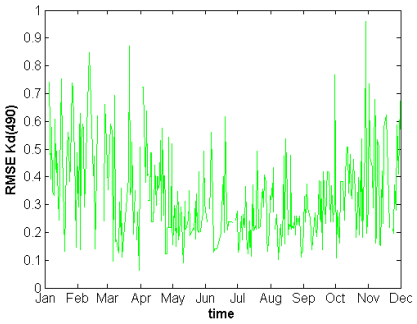
The performance of the filter will be investigated during the months March, April and May, belonging to the spring bloom. The system noise on the state variables need to be defined in such a way, that it will coincide with the noise that is required. The magnitude of the relative error that is needed will be investigated, using the months March, April and May. The system noise needs to be such that the ensemble of model states will overlap with the ensemble of observations. Therefore, in figure 41 we show the relative error that the model makes when compared to the MERIS data. The relative error is defined as $\frac{|model-meris|}{model}$. The median of this relative error is shown, calculated using the months March, April and May, for both the *Chlfa* and Kd. In some cases, the relative error, turned out to be very large. Unrealistic values of a relative error of much greater than one were calculated, but these values were ignored. These values might be due to a bias occurring in the difference between the model and the MERIS data.

Since the relative error of the model in the filter is defined to be constant over the grid, we calculated the overall median value. For *Chlfa* this value is approximately 0.4 and for K_d it is approximately 0.3. Recall that in the twin experiments in the previous chapter, the relative measurement noise that was used, was equal to the relative system noise that was used. Therefore, we decided to calculate the relative measurement noise, given by the MERIS data as well. This value for *Chlfa* is given in figure 42. Here we see that the relative measurement noise of the *Chlfa* is smaller along the coast and larger in the open sea. It seems that the measurement noise is much smaller than the required system noise.

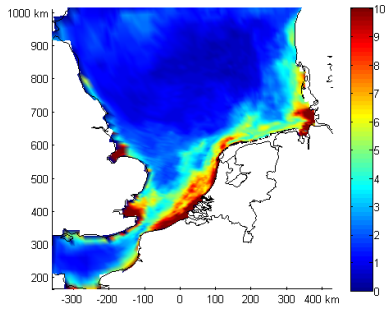
In figure 43 the average value of *Chlfa* is shown for four different OSPAR regions given by the MERIS data. OSPAR regions are defined by the OSPAR



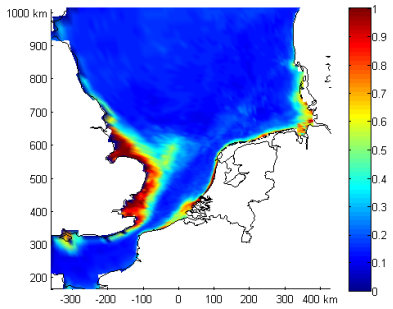
(a) Timeseries RMSE *Chlfa*



(b) Timeseries RMSE K_d

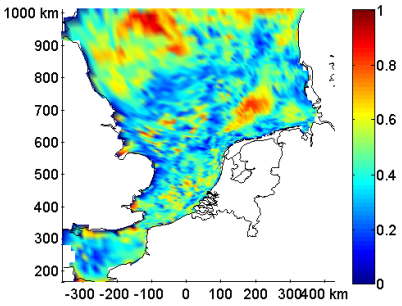


(c) Spatial RMSE *Chlfa*

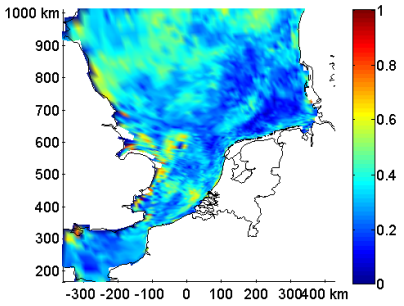


(d) Spatial RMSE K_d

Figure 40: ***Spatial and time series RMSE of Chlfa and Kd. Comparing the model without system noise with the MERIS data.***



(a) Median relative error *Chlfa*



(b) Median relative error K_d

Figure 41: ***Median of the relative error that the model makes compared to MERIS. Median calculated over the months March, April, and May. Relative error defined as*** $\frac{|model-meris|}{model}$***, only values smaller than one were taking into account in calculating the median. Medians are shown for the Chlfa and*** K_d***.***

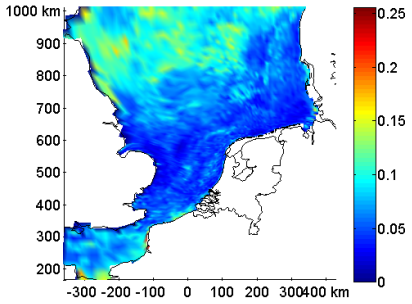


Figure 42: ***Median of the relative error of Chlfa given by MERIS. Median calculated over the months March, April, and May. Relative error calculated as*** $\frac{std}{chlfa}$.

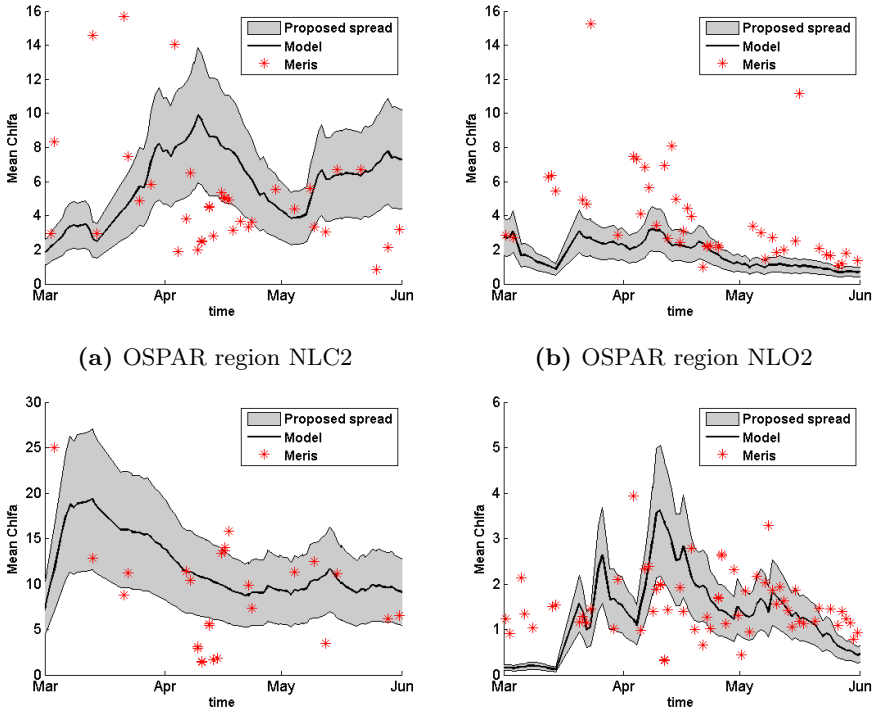
convention and shown in figure 70 in the appendix. Also shown is the modelled *Chlfa* with an error of 40%. It seems that in the filter, a relative error on *Chlfa* of at least 0.4 is indeed needed. Therefore, in the experiments, we start with a relative error of 0.6 on the state variables, such that after one day, the relative error in the *Chlfa* forecast is at least 0.4.

As mentioned before, this model error is much larger than the measurement errors given by the MERIS data. This shows to be a problem for the filter behavior. Therefore, the measurement noise is increased. To deal with other problems that arise, also other additional features are added to the filter, which will be explained now.

11.2 Added features

MERIS data is available almost daily, at times 11:00 and 12:00. To make the filter more general, measurement updates will take place at 12:00 only. Measurements at 11:00 will be used as measurements at 12:00, and measurements that were taken on a single day at both 11:00 and 12:00 will be combined. For testing purposes, the ESRF without localization, was directly applied to this MERIS data. Initially, there were some problems with the filter, which were attempted to be solved by adding several features.

One of the problems was that when the filter forecasts an almost zero *Chlfa*, the filter is not able to update and move upwards to a nonzero *Chlfa* value given by the MERIS data. This is because the model noise is very small when a substance is very small, because we defined a relative error. To solve this, an extra system noise is added, to make sure that the model noise is large enough in these type of places. It was decided to add noise with a constant standard deviation to each of the substances separately. For each substance, the mean value over the whole grid is calculated and a fraction of this mean value is taken as the standard deviation for the extra noise on this substance. For instance, we can take 50% of the mean value. Consequently, at places with a small model noise, the model noise becomes larger, while at other places this extra noise does not have a big impact.



(a) OSPAR region NLC2

(b) OSPAR region NLO2

(c) OSPAR region NLWD

(d) OSPAR region UKO5

Figure 43: ***Mean Chlfa values for four different OSPAR regions. Shown is the model, with a spread showing the proposed relative error of 0.4 in Chlfa. Also shown are the values given by MERIS.***

When adding this extra noise, it often occurs that some state variables become negative. These values are not corrected, such that no bias is introduced. Correcting occurs after the measurement update. All mean values that are negative are set to zero and the ensemble perturbations are decreased with a constant factor, until all ensembles are positive. This approach makes sure that the mean value is not changed, only the spread of the ensemble is adapted. Such that, again, no bias is introduced.

It was noted that adding this extra noise did not completely solve the problem of divergence. Still, the filter behaved strangely, very unpredictable. At most places, the MERIS data has a very small uncertainty, but the filter did not converge to this value. Looking back at chapter 8, we decided to increase the standard deviation of the MERIS data, by multiplying it with a factor, for instance 4. This indeed solved the problem.

After a couple of test runs, it turns out that on certain days, still, the filter diverges. It seems that this happens only when there are a small number of measurements given by MERIS. For this reason, we decided to apply localization as described in chapter 9 and in the twin experiments.

11.3 Results

In [35] it was recommended to start assimilating in March, as in the months before March, there seems to be a bias between the model and the MERIS measurements. We therefore start on the first day of March, where we start with a large number of MERIS observations. The relative error for the model without localization was set to 0.6, such that after a model run, we have approximately a relative error of 0.4 in the *Chlfa*. Furthermore, we add 100%·mean as extra noise on each substance separately. Localization is applied and we multiply the measurement noise by 4. Validating takes place with *in situ* measurements.

In table 12 and figure 44 we show all RMSE values. Here we see that overall, the assimilated MERIS data improves the forecast a bit. The time series shows that the forecast is improved almost every time (when there are *in situ* measurement available. However, not at every segment the forecast is improved, as sometimes the RMSE increases.

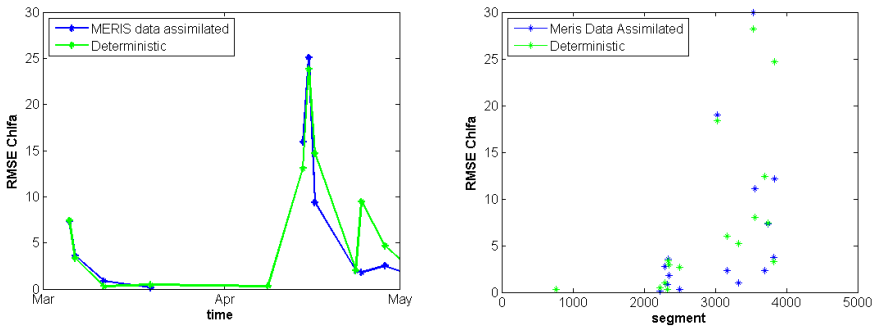
To give an idea of the workings of the filter, in figure 45 we show examples of filter updates on March 1 and March 2. Here it seems that the filter works quite well. The filter converges to the MERIS data on March 1, but due to the localization, the filter stays close to the model as well. However, there still seems to be some strange behavior visible on March 2. It is not clear what the cause is of this behavior and if it is actually wrong. Overall, the filter seems to perform well in the first days of assimilating. However, when we evaluate the filter on days 23 April and 24 April, in figure 46, the filter does not seem to perform that well. It seems that an ensemble collapse occurred and also the filter gives a non-smooth unpredictable solution. This is more evident when we view the time series of the *Chlfa* in segment 3556, which is station NZR6NW010, shown in figure 47.

We would also like to point out, that in the beginning of the assimilation period, the filter behaves quite well, but after a certain period it was noted that

the filter diverges to unrealistic high values. The reason for this is not clear. For now, the most likely reason seems to be that the model error is greatly overestimated and therefore not only an ensemble collapse occurs, but the update also gives unrealistic values in some segments. There was unfortunately no time to investigate this properly. A number of other parameters were investigated to help improve the filter, to no avail. For instance, the local regions where increased and the system noise was decreased, but the results were very similar or worse. Note that the increase of the measurement noise resulted in negative measurements of *Chlfa*, this was not corrected for in this thesis, but is recommended.

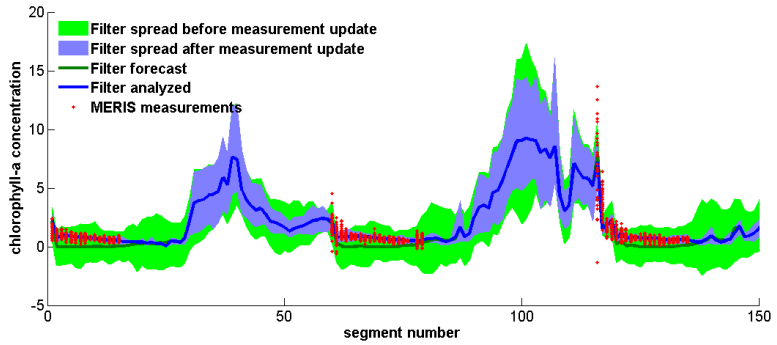
	Deterministic	DA
Chlfa	11.179	10.5198

Table 12: ***Overall RMSE of Chlfa. Comparing the model comparing model with assimilation of MERIS.***

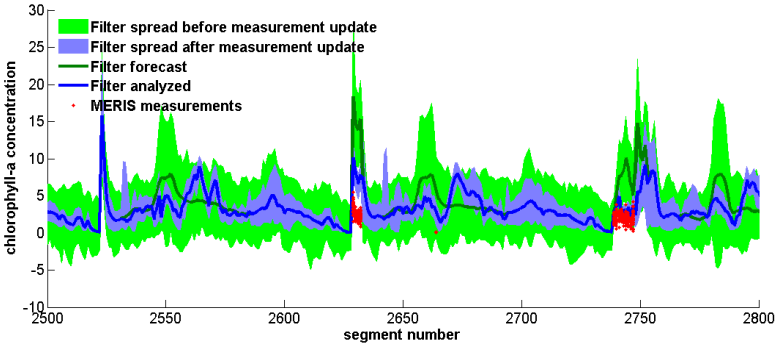


(a) Timeseries RMSE **(b)** Spatial RMSE

Figure 44: ***Timeseries and spatial RMSE, comparing model with assimilation of MERIS.***

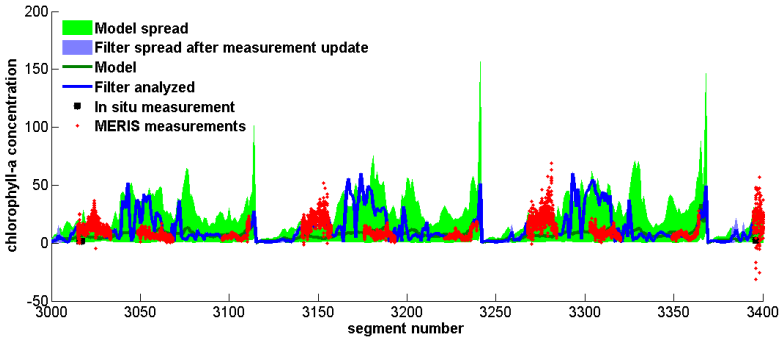


(a) March 1

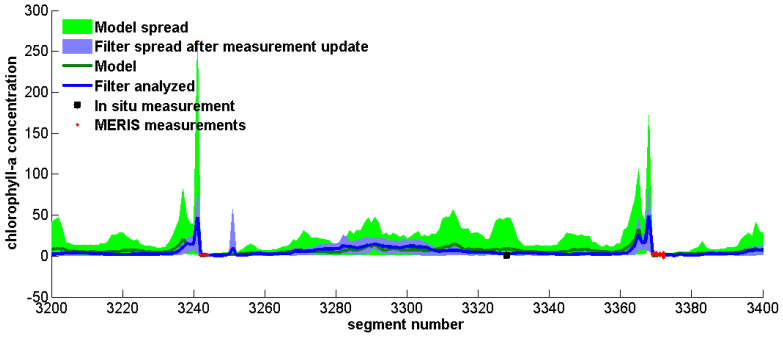


(b) March 2

Figure 45: ***Results of assimilating MERIS data on March 1 and 2. Shown are a number of segments.***



(a) April 23



(b) April 24

Figure 46: ***Results of assimilating MERIS data on April 23 and April 24. Shown are a number of segments around the place where there is an in situ measurement.***

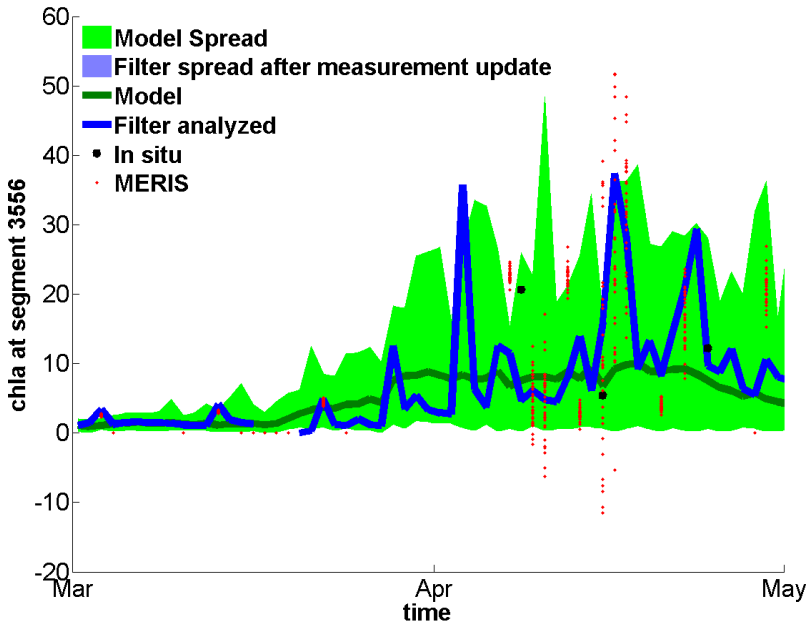


Figure 47: ***Chlfa time series at segment 3556, station NZR6NW010.***

12 Discussion

This chapter discusses the chosen methods. We reflect critically on the work that was done and this provides an outlook for future work, as described in the Recommendations.

For the system noise, that represents the uncertainty of the model, a relative error was defined and used in the twin experiments. However, the experiments on the MERIS data showed that another error (an additive one) on the substances was needed. This was not investigated in the twin experiments, so it is not clear what the magnitude of this extra additive noise should be and what the effect on the model and the data assimilation is. In the description of the system noise it was assumed that the substances were uncorrelated, which is likely invalid.

The relative error was modelled in two different ways. The first method defined a squared exponential correlation function with a certain correlation scale and generated noise by approximating the square root of the resulting covariance matrix. The covariance matrix was approximated by setting several values to zero, such that it could be saved into the memory. This was however not done in a consistent way, such that the resulting covariance matrix turned out to be not positive definite. Consequently, it is not very clear how the singular value decomposition that lead to the approximation of the square root matrix of the covariance matrix behaves.

The other method of generating system noise relies on interpolation of white noise. The method of Kriging was investigated, as to construct a squared exponential correlation, but this turned out to be hard to tune, as a regularization constant was required. So, interpolation weights were chosen by evaluating the resulting correlation function on a 1D grid example. Only two different weights were investigated, inverse distance and exponential. And the exponential weight was chosen, based on that the resulting correlation was smoother and more similar to the squared exponential.

Covariance localization methods were investigated. The one that was chosen was domain localization for the ESRF, as this was the method that was most easily applicable. Other methods turned out not to work, or require an enormous amounts of computation time. However, also the domain localization had a problem. At some segments, the filter diverged catastrophically. This is most likely due to a small number of measurements and small measurement noise. No general threshold was found. In order to still be able to apply this localization, the ESRF implementation was rewritten and bounding was applied. It has shortly been investigated how sensitive the filter is to this bound, and it turned out not to make a big difference. However, it is still not yet completely clear what the dynamics are of this localization with bounds.

The rewritten ESRF that was used for covariance localization is actually a better implementation for the ESRF, and should have been used initially. This rewritten implementation makes sure that the ensemble perturbations of the measurements are truly not used and is possibly an improvement as it will be less sensitive to sampling error in measurements.

In order to define the measurement noise that must be used in the twin experiments, the propagation of noise in the model was investigated. Since we

wanted to assume a constant measurement noise in space and time, the median was evaluated. The noise was relatively constant in space, but it increased in time. So this assumption might not be valid.

Furthermore, it was not investigated if the error on the substances is actually normally distributed, which is an assumption for the data assimilation in this thesis. Also, it was not investigated if a normal distributed noise on the substances resulted in a normally distributed noise on the *Chlfa* when propagated by the model. The EnKF and ESRF assume normally distributed errors and when this is not the case, the filter may not perform well as the error are described in a wrong way.

In the twin experiments validation took place by evaluating several RMSE values. In most cases, the RMSE were very similar and sometimes it was hard to make a decision on which filter was better. RMSE values of one filter may be smaller than the ones resulting from another filter, but if this is significant is a different question. Usually, this is investigated by performing the filters a multiple time, as to reduce a random effect. This was however not done, because of time constraints.

Also, the statistics of the innovations were investigated in the twin experiments. The percentages between σ bounds were evaluated. These statistics were calculated for all segments. It is not yet clear how well each filter works along the whole grid and why in some segments there is a good correspondence with the theoretical statistics and in some segment there is not. Every filter performs different spatially. Only the average along the grid was evaluated and only the innovations of two segments were more thoroughly examined.

The filters in the twin experiments only used measurements on the *Chlfa*. Therefore, only the innovation statistics for the *Chlfa* could be evaluated. The RMSE, however, was also evaluated for the other variable of interest, the K_d. The substances, the state variables themselves were however not evaluated. This can be done, since the true state is completely known. Now it is not yet completely clear how well the measurement update behaves for the state variables and if they remain realistic, biologically.

The localization method gave a lot of incorrect statistics for the innovations and was improved by decreasing the system noise. Localization also showed to improve when using larger local regions. Initially, we choose the local regions as large as the correlation scales defined for the system noise. Since this localization method works better with larger local regions, it might be that the correlation scale for the system noise can be larger as well. Note that the correlation scale that was used, resulted from previous experiences at Deltares. Other correlation scales were however not investigated in this thesis.

Now, in the localized ESRF, only the first two layers of the North Sea were updated using the measurement on the surface. This may have let to a very non-smooth solution, which was not examined in this thesis. Only the performance of the filter on the water surface was evaluated.

When the filter was applied to the MERIS application, some additional features were added. For instance, the standard deviation of the observations given by MERIS were increased, which is maybe a bit odd, as the standard

deviation is given and should be used accordingly. Furthermore, it is not yet clear how these additional features affect the performance of the filter. The filter in general seems to work well, however, it does not yet improve the fit with the *in situ* measurements at all segments. Furthermore, the filter behaves strangely at some points in a later period of assimilating, which seems like an ensemble collapse with filter divergence in some points. So, the filter does not yet work optimally and therefore it is hard to say if the filter actually improves the forecast.

13 Conclusion

This chapter summarizes and concludes the results that were obtained in this thesis. The results obtained from the twin experiments, which investigated the filters, are described, as well as the filter applied to the MERIS data.

The EnKF was evaluated first. In general, the filter showed to perform well. So, the chosen system noise and measurement noise seem to be alright. The RMSE values of the *Chlfa* and K_d are greatly reduced, compared to the determinstic model. Not at all segments this was the case, since at some segments there is a small increase of the RMSE. On average, the statistics of the innovations are close to the theory. Again, however, not at every segment the statistics of the innovations are correct. At some segments, there is a filter divergence from the truth, at other segments an ensemble collapse occurs.

The ESRF and EnKF perform similarly, there is only a small difference in the RMSE values and innovation statistics spatially. At some segments the EnKF works better and at other segments the ESRF performs better. The ESRF is known to be more numerically stable and this also is the case in the twin experiment, since the filter diverges less from the truth. Therefore, the ESRF is preferred.

The two methods of generating system noise were compared by applying them to the ESRF. Again, no substantial difference is found in the RMSE values. The type of system noise are hard to compare, since the results differ spatially. The interpolated noise, however, seems to introduce some large RMSE values on the grid. Therefore, the SVD generated noise is preferred.

Next, the localization method was applied to the ESRF. The RMSE values were greatly reduced, but this result turned out to be invalid, since the statistics of the innovations did not correspond by what is known theoretically. Now at a lot of segment, an ensemble collapse occurred. Therefore, two adapted localization methods were applied. The first decreased the system noise and the second increased the local regions. Both methods made sure that the measurements were not taken into account that much in the update, to avoid ensemble collapse. In both cases, the innovations corresponded better to the theory and the RMSE was greatly reduced.

The effect of increasing the ensemble size was investigated on the ESRF without localization. Increasing the ensemble size from 30 to 50 to 100 subsequently improved the prediction of the ESRF. The difference between 30 and 50 ensembles is not very profound. The ensemble size 30 is chosen as it a practical number for which the ESRF works well and it is more efficient to apply localization to improve the filter, than to increase the ensemble size.

In all cases, the improvements of predictions made by filters are more profound on the *Chlfa*, as these values are used in the data assimilation. The K_d prediction can also be improved by improving the filter, but not as much.

It is needed to add additional features to the filter when applying it to the MERIS data. An extra additive noise is needed on the state variables, to be able to deal with small *Chlfa* concentration forecasted by the filter, that are overestimated by definition of a relative error. Since the filter applied to the MERIS data is less controlled than in the twin experiments, negative values

of substance occur. Those are dealt with by decreasing the ensemble variance after a measurement update, such that no bias is introduced. Furthermore, it was needed to increase the standard deviation of the measurement noise given by MERIS. Still, filter divergence occurs when only a small number of MERIS measurements are available. Applying localization solves this problem. The filter now seems to work quite well, however, the resulting solution does not fit the *in situ* data at all points and the filter can behave strangely still, it looks like ensemble collapse occurs often in a later stage in the assimilation period, with some filter divergence behavior. The filter has to be tuned greatly before it can be put in use.

14 Recommendations

This chapter offers some recommendations that will be useful for future work. These recommendations are largely based on the discussion.

With respect to the system noise generated by the SVD of the squared exponential correlation matrix, we recommend to adjust the way of saving the matrix into memory. Now all values below 0.5 are set to zero, which results in a matrix that is no longer positive definite. It is wiser to increase the number of block or diagonals in this matrix, until adding new block or diagonals do no longer fit in the memory. This makes sure that the matrix is still positive definite and more consistent. Possibly, the SVD of this matrix behaves better.

It is also possible to approximate the SVD generated noise more accurate, by using the fact that the squared exponential correlation function is separable in X, Y and Z. For each direction in space, a square root of the matrix can be calculated. This makes the computation much more efficient and accurate. This perhaps makes it possible to easily investigate different correlation scales or even correlation functions, which we recommend.

Also, the other method of generating noise by interpolating white noise can be investigated more thoroughly, by researching different weights. Other methods for generating noise are available as well in the literature, which can be examined. We recommend to investigate cross-correlated substances, as the errors in substances are likely to be related.

We also strongly recommend to investigate the noise propagation in the model more thoroughly, especially now that we know that an extra additive noise is needed, besides a relative error. Also, the Gaussian assumption can perhaps be validated. With respect to the measurement noise, it is wise to investigate a time dependent measurement noise in the twin experiments. This might improve the results.

Furthermore, we strongly recommend to include the K_d in the data assimilation and also the parameters. Parameters might go out of physical bounds, so we recommend to add constraint in the data assimilation. Possibly by transforming the parameters to an infinite space, using for instance the log function and after the update transform them back.

We also recommend to investigate the filter using the twin experiments more thoroughly. It is not yet clear how the filter works in different regions. In some segments the filter works perfectly, at other it diverges, either from the truth or there is an ensemble collapse. It is recommended to investigate how to make this more consistent. Also, it is wise to investigate the effects of the update on the state variables. See if the effects are realistic, biologically speaking.

Another recommendations is to rewrite the implementation of the ESRF, by using the inverse of $\mathbf{R}$, like in the localization method. This makes sure that no samples of the measurement noise are used and might give more insight in the way the filter performs. With respect to the localization, we recommend to investigate this more thoroughly, since it requires tuning. Furthermore, in each local region, all measurements are considered equally by the filter. It is possible to increase the measurement noise farther away from the segment of interest,

such to accomplish a more smooth localization. In this way, we possibly update more layers, more smoothly.

There is a filter, named the Deterministic Ensemble Kalman Filter (DEnKF) [29] that combines the EnKF and ESRF, resulting in a filter that is as stable as the ESRF but has the advantages of a simple implementation of the EnKF, such that a Schur product localization can easily be applied. However, if one applies this filter, parallel computation need to be investigated, since at this point applying the Schur Product to the EnKF requires a ridiculous amount of computation time.

With respect to the application to the MERIS data. In [35] it is suggested to investigate the use of a bias aware filter [7], since there seems to be a bias between the model and the MERIS data. Furthermore, we recommend to investigate the effect of a small number of measurements more thoroughly, since a problem seems to arise here. Finally, it is very important to tune the filter and investigate the effect of all parameters, including the size and time dependence of the system noise.

15 References

References

[1] http://ices.dk/.

[2] A. Barbu. *Ensemble-based data assimilation schemes for athmospheric chemistry models*. PhD thesis, Technische Universiteit Delft, 2010.

[3] M Blaas, K Cronin, G.Y. El Serafy, Y.F. Friocourt, I.D.T.F Garcia Triana, S Gaytan Aguilar, and G.H Keetels. Mos2: Model setup, data assimilation and skill assessment. Technical report, Deltares, 2012.

[4] A.N. Blauw, F.J. Los, M Bokhorst, and P.L. Erftemeijer. Gem: a generic ecological model for estuaries and coastal waters. *Hydrobiologia*, 618:175–198, 2009.

[5] NOAA Coastwatch. K490 - diffuse attenuation coefficient at 490 nm.

[6] S Dance, R Petrie, N Nichols, and M. Butala. Stability and localization in the ensemble square root filter, 2009.

[7] D.P. Dee. Bias and data assimilation. *Q.J.R. Meterological Society*, 131:3323–3343, 2005.

[8] Deltares. In vivo chlorophyll fluorescence measurements. Technical report, Deltares, 2012.

[9] M.A. Eleveld, H.J van der Woerd, and H. Beck. Improving the estimation of north sea primary production: Meris chl and kd in vgpm. *Envisat Symposium 2007*, 2007.

[10] G. Evensen. The ensemble kalman filter: theoretical formulation and practical implementation. *Ocean Dynamics*, 53:343–367, 2003.

[11] G. Evensen. Sampling strategies and square root analysis schemes for the enkf. *Oceam Dynamics*, 54:539–560, 2004.

[12] G.A. Gottwald and A.J. Majda. A mechanism for catastrophic filter divergence in data assimilation for sparse observation networks. *Nonlinear Processes in Geophysics*, submitted, 2012.

[13] N. Halko, P.G. Martinson, and J.A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *Society for Industrial and Applied Mathematics REVIEW*, 53:217–288, 2011.

[14] J. Harlim and A.J. Majda. Catastrophic filter divergence in filtering nonlinear dissipative systems. *Community of Mathematical Sciences*, 8:27–43, 2010.

[15] P.L. Houtekamer and H.L. Mitchell. A sequential ensemble kalman filter for atmospheric data assimilation. *American Meteorological Society*, 129:123–137, 2000.

[16] B.R. Hunt, E.J Kostelich, and I. Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform kalman filter. *Elsevier Physica D*, 230:112–126, 2007.

[17] R.E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME Journal of Basic Engineering*, 82:35–45, 1960.

[18] H Li, M Arias, A Blauw, H Los, A.E. Mynett, and S. Peters. Enhancing generic ecological model for short-term prediction of southern north sea algal dynamics with remote sensing images. *Ecological Modelling*, 221:2435–2446, 2010.

[19] F.J. Los, M.T. Villars, and M.W.M. Van der Tol. A 3-dimensional primary production model (bloom/gem) and its applications to the (southern) north sea (coupled physical-chemical-ecological model). *Journal of Marine Systems*, 74:259–294, 2008.

[20] H. Los. Mathematical simulation of algael blooms by the model bloom ii. Technical report, Delft hydraulics laboratory, 1991.

[21] H. Los. *Eco-hydrodynamic modelling of primary production in coastal waters and lakes using BLOOM*. PhD thesis, Wageningen University, 2009.

[22] R Niu, P.K. Varshneu, M Alford, A Bubalo, E Jones, and M. Scalzo. Curvature nonlinearity measure and filter divergence detector for nonlinear tracking problems. Technical report, Syracuse University New York and Air Force Research Labortatory Rome, 2009.

[23] OurLake. Data interpretation: Chlorophyll, September 2009.

[24] L. Peperzak. Climate change and harmful algal blooms in the north sea. *Acta Oecologica*, 24:139–144, 2003.

[25] L Peperzak and M. Poelman. Mass mussel mortality in the netherlands after a bloom of phaeocystis globosa (preymnesiophyceae). *Journal of Sea Research*, 60:220–222, 2008.

[26] R.E. Petrie. Localization in the ensemble kalman filter. Master's thesis, Department of Meteorology, University of Reading, 2008.

[27] EC-FP5 REVAMP project. Atlas of chlorophyll-a concentration of the north sea based on meris imagery of 2003, May 2005.

[28] P Sakov and P.R. Oke. Implications of the form of the ensemble transformation in the ensemble square root filters. *Monthly weather review*, 136:1042–1053, 2007.

[29] P Sakov and P.R. Oke. A deterministic formulation of the ensemble kalman filter: an alternative to ensemble square root filters. *Tellus*, 60A:361–371, 2008.

[30] K Salacinska, G.Y. El Serafy, F.J. Los, and A. Blauw. Sensitivity analysis of the two dimensional application of the generic ecological model (gem) to algal bloom prediction in the north sea. *Ecological Modelling*, 221:178–190, 2010.

[31] G.Q. Tabios and J.D. Salas. A comparative analysis of techniques for spatial interpolation of precipitation. *Water Resources Bulletin*, 21:365–380, 1985.

[32] N.A. Thacker and A.J. Lacey. Tutorial: The kalman filter. Technical report, Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester, 1998.

[33] H.J Van Der Woerd and R. Pasterkamp. Hydropt: A fast and flexible method to retrieve chlorophyll-a from multispectral satellite observations of optically complex coastal waters. *Remote Sensing of Environment*, 112:1795–1807, 2008.

[34] X Yang and T. Delsole. Using the ensemble kalman filter to estimate multiplicative model parameters. *Tellus*, 61A:601–609, 2009.

[35] D. Zhang. Improving the algal bloom prediction on the north sea by dual ensemble kalman filter in gem model. Master's thesis, Department of Applied Mathematics Delft University of Technology, 2012.

[36] D Zhang. Statistical analysis by using earth observation in diffuse attenuation coefficient. Technical report, Deltares, 2012.

16 Appendix

16.1 Extra figures

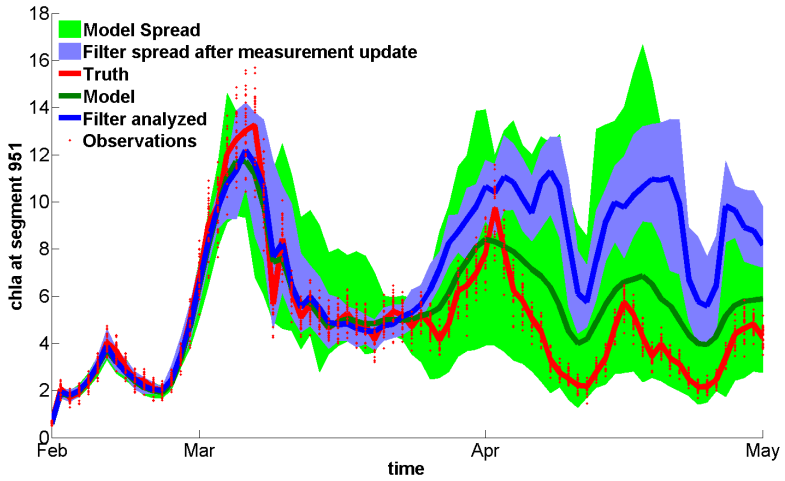


Figure 48: ***This figure shows the model state and EnKF state of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

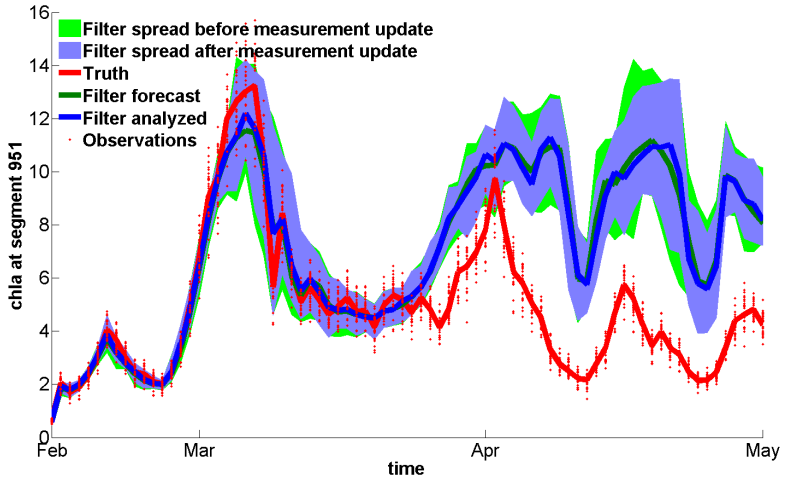


Figure 49: ***This figure shows states of the EnKF before and after measurement update, of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

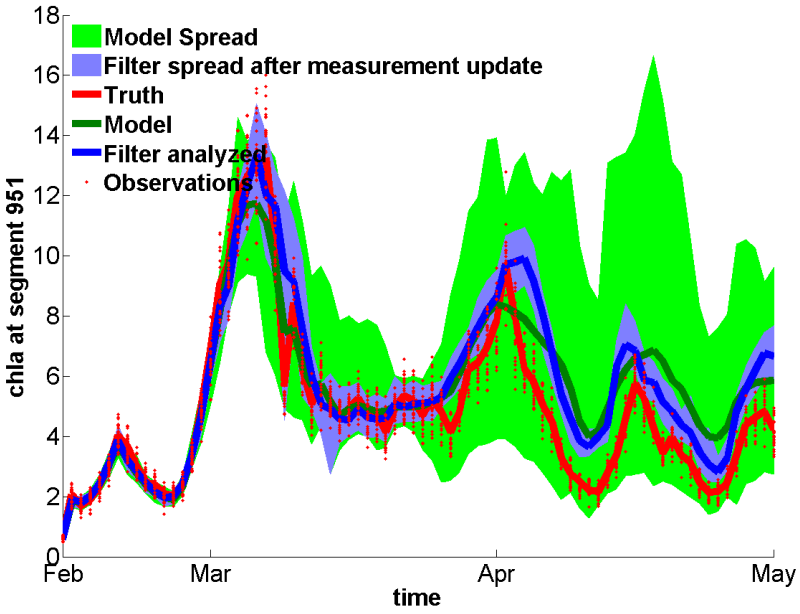


Figure 50: ***This figure shows the model state and ESRF state of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

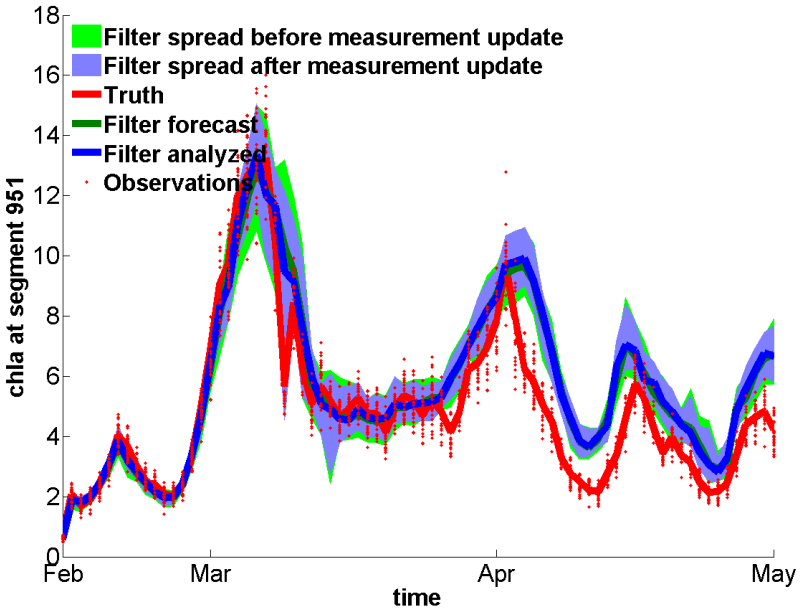


Figure 51: ***This figure shows states of the ESRF before and after measurement update, of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

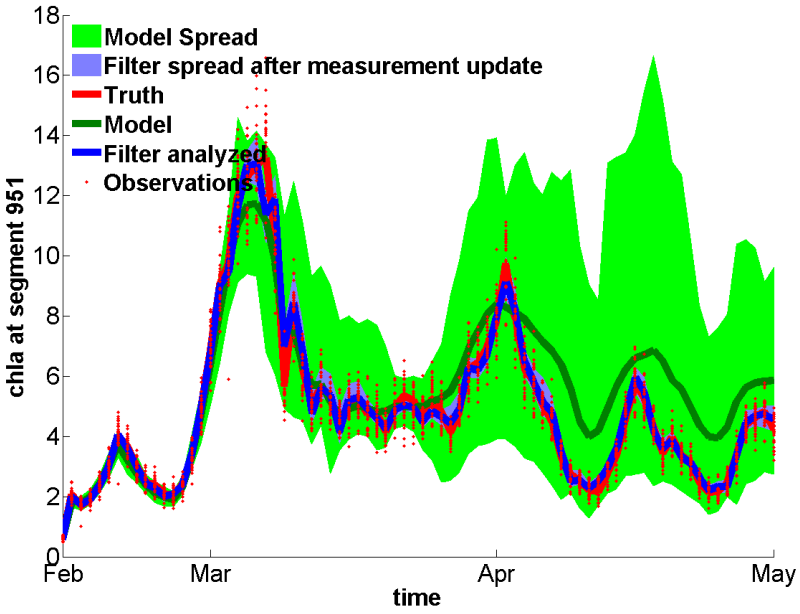


Figure 52: ***This figure shows the model state and localized ESRF state of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

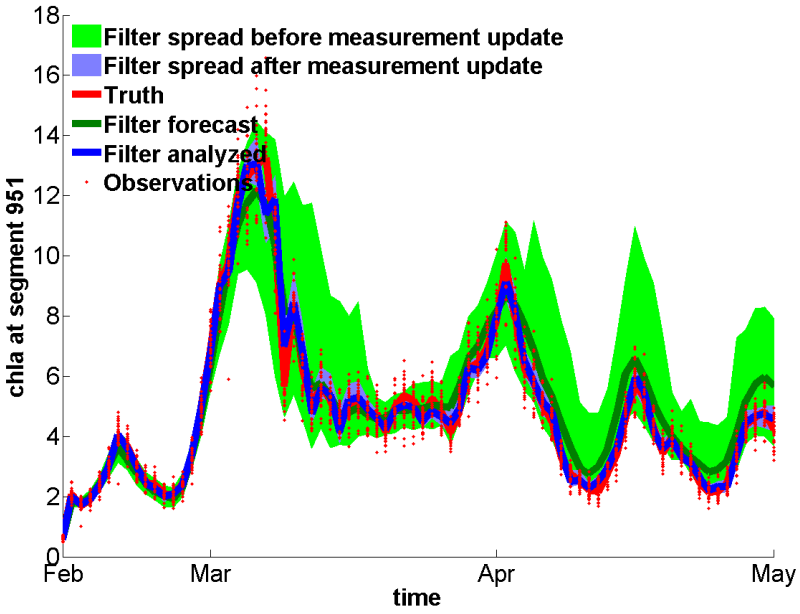


Figure 53: ***This figure shows states of the localized ESRF before and after measurement update, of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

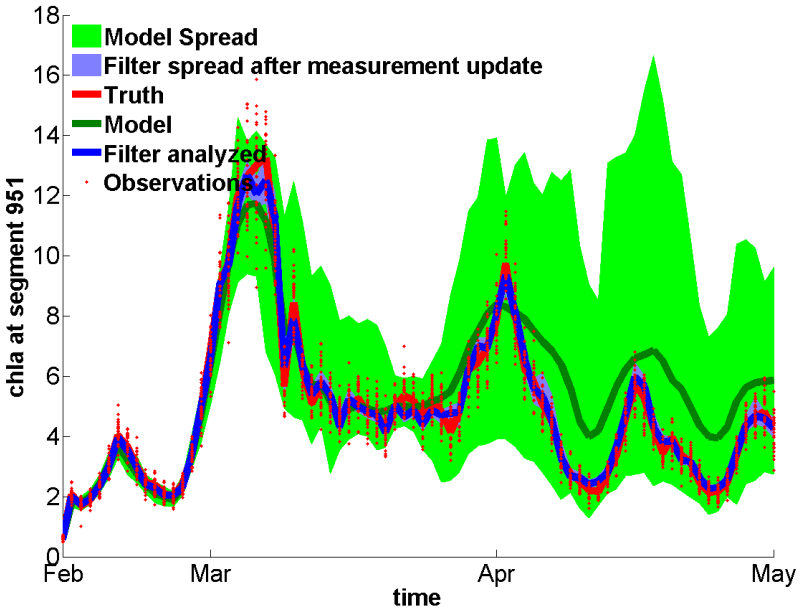


Figure 54: ***This figure shows the model state and localized ESRF 2 state of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

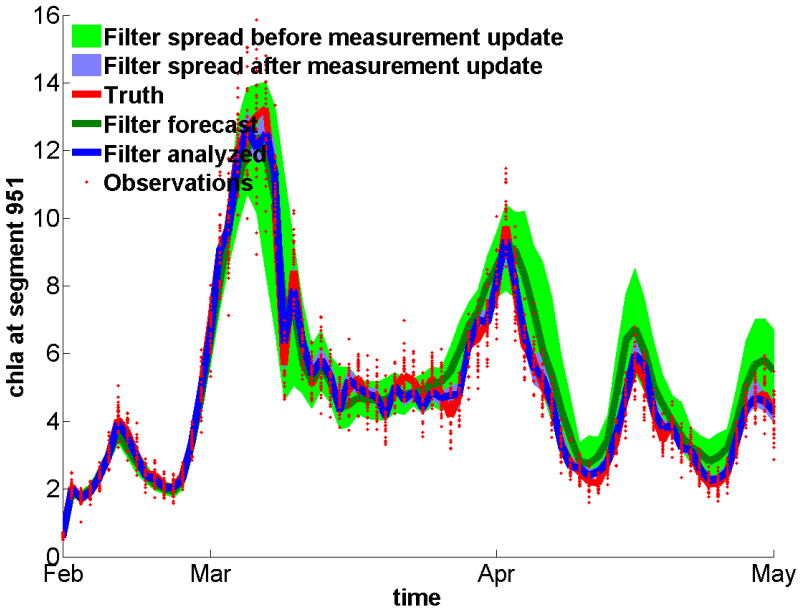


Figure 55: ***This figure shows states of the localized ESRF 2 before and after measurement update, of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

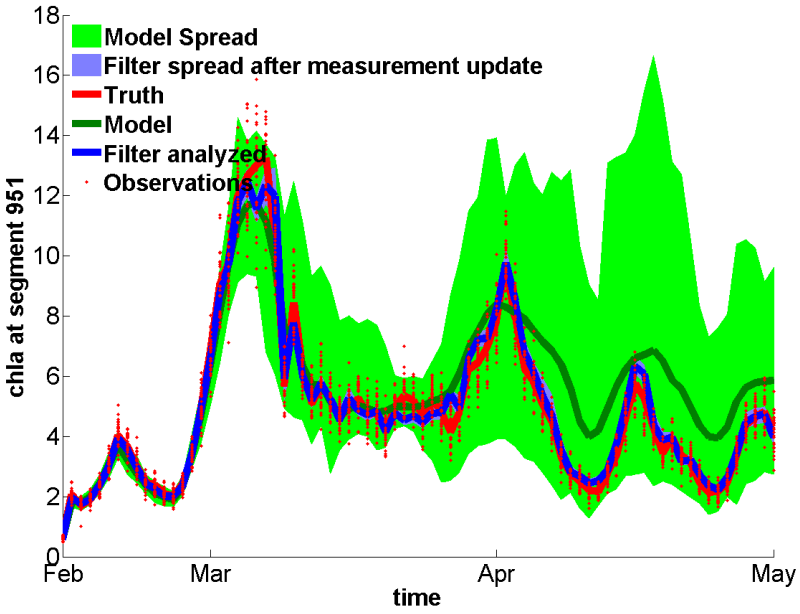


Figure 56: ***This figure shows the model state and localized ESRF 3 state of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

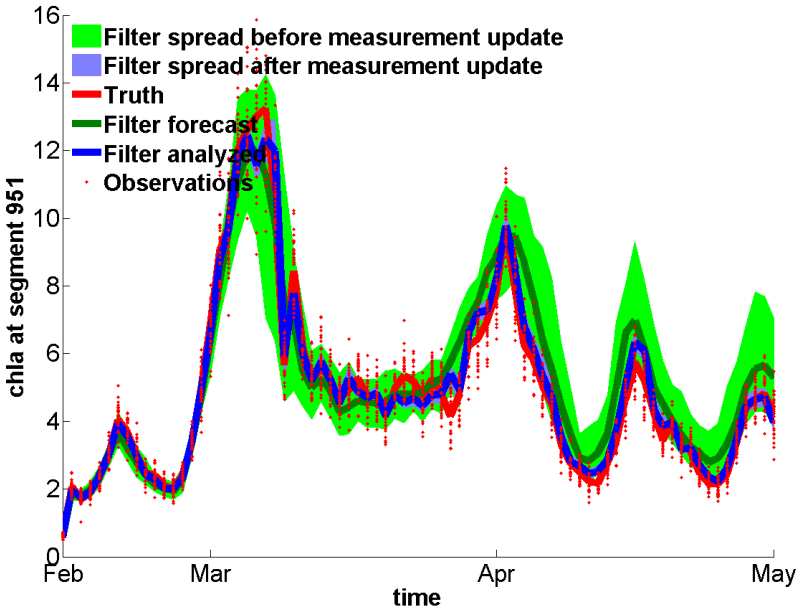


Figure 57: ***This figure shows states of the localized ESRF 3 before and after measurement update, of the Chlfa concentration at segment 951. Also shown is the true state with observation noise.***

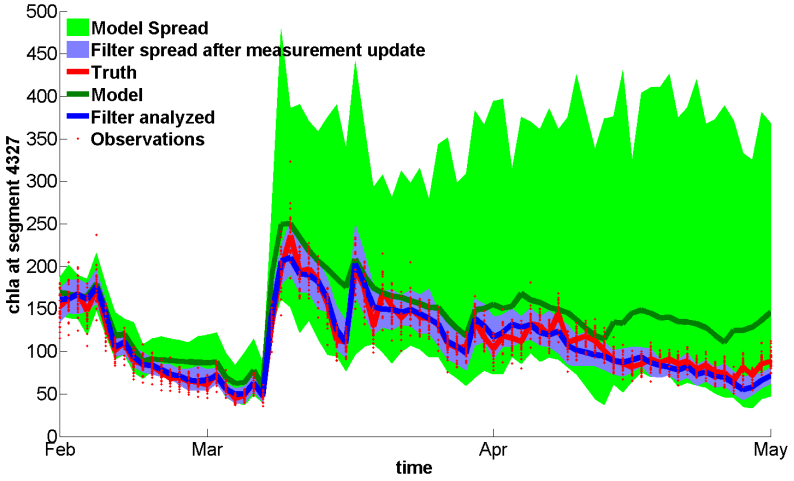


Figure 58: ***This figure shows the model state and EnKF state of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

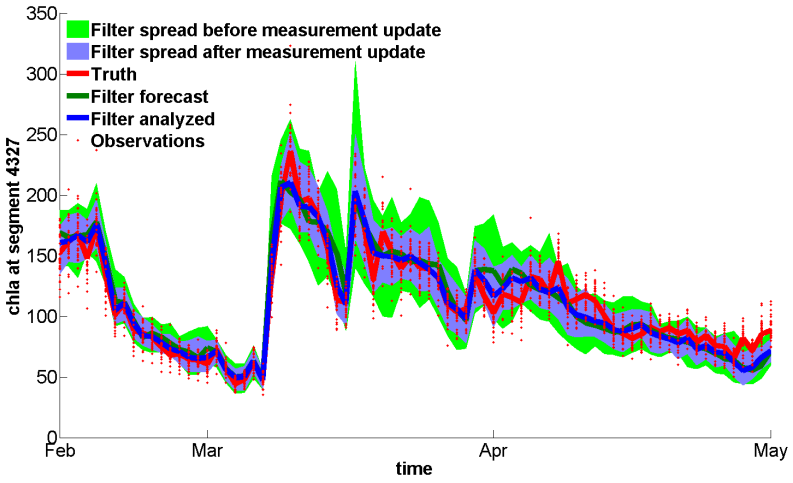


Figure 59: ***This figure shows states of the EnKF before and after measurement update, of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

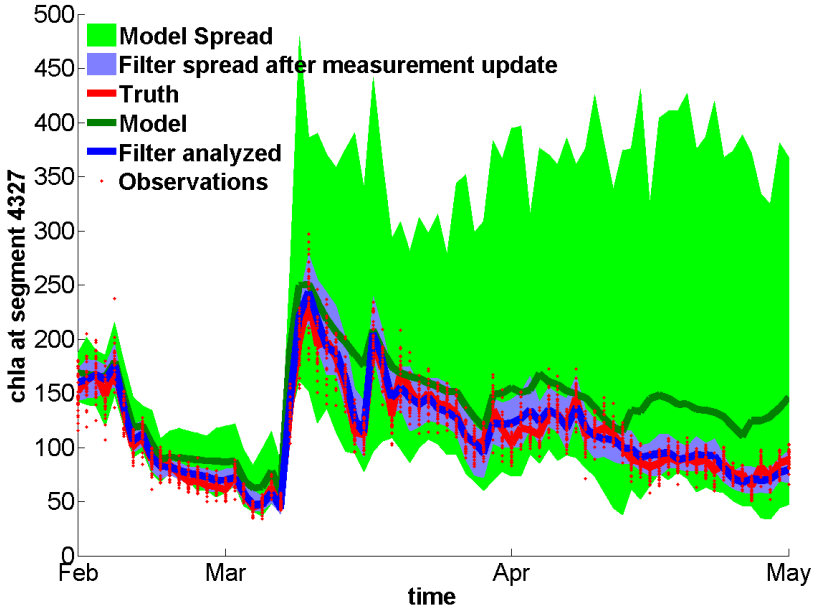


Figure 60: ***This figure shows the model state and ESRF state of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

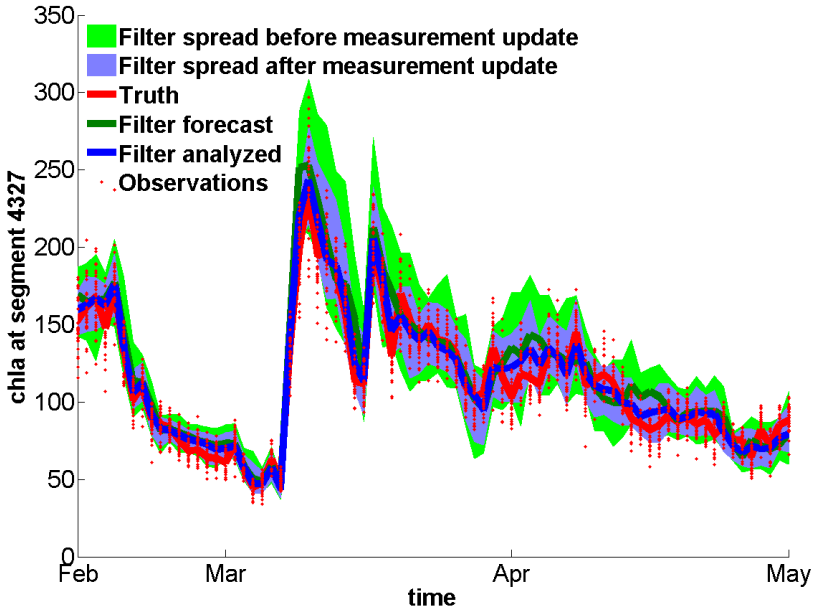


Figure 61: ***This figure shows states of the ESRF before and after measurement update, of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

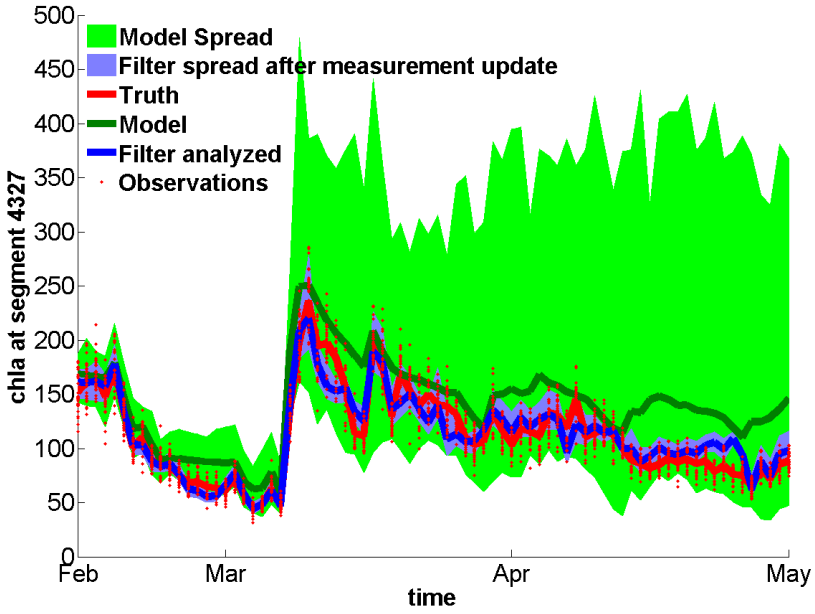


Figure 62: ***This figure shows the model state and localized ESRF state of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

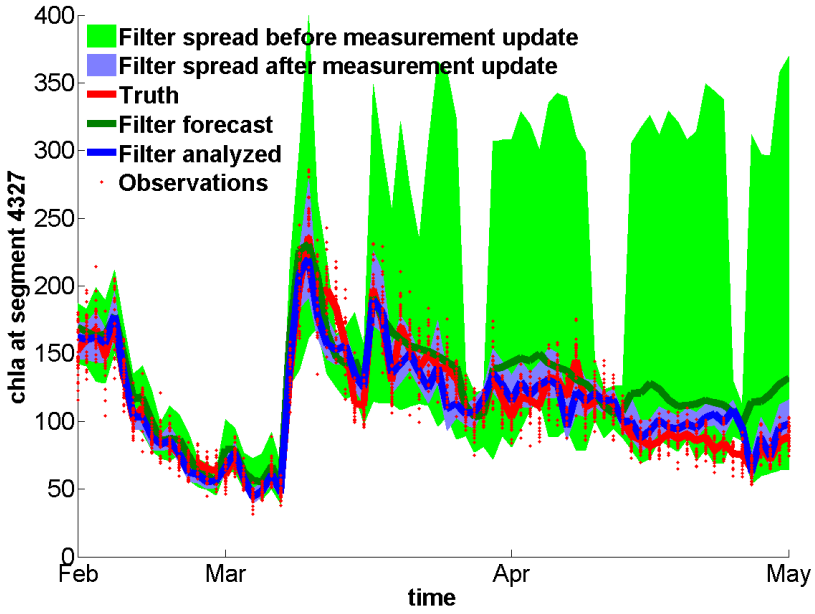


Figure 63: ***This figure shows states of the localized ESRF before and after measurement update, of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

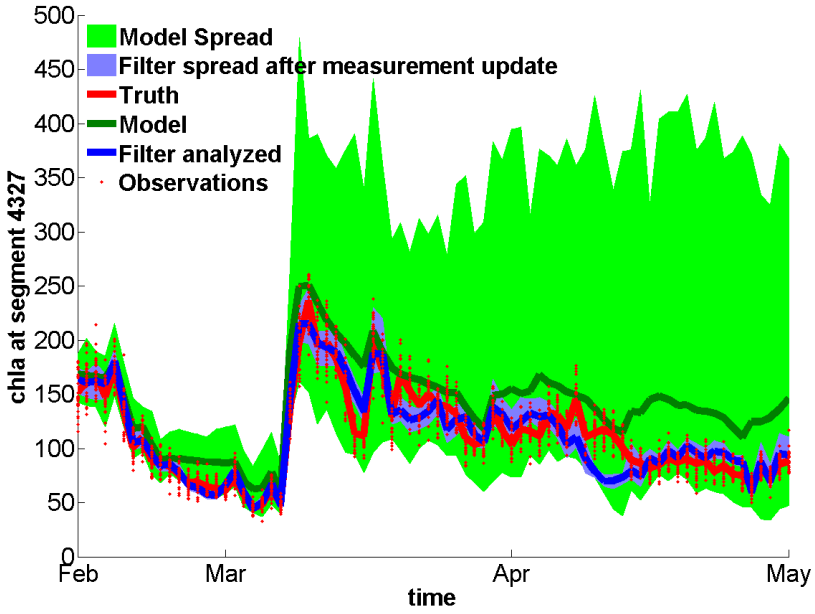


Figure 64: ***This figure shows the model state and localized ESRF 2 state of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

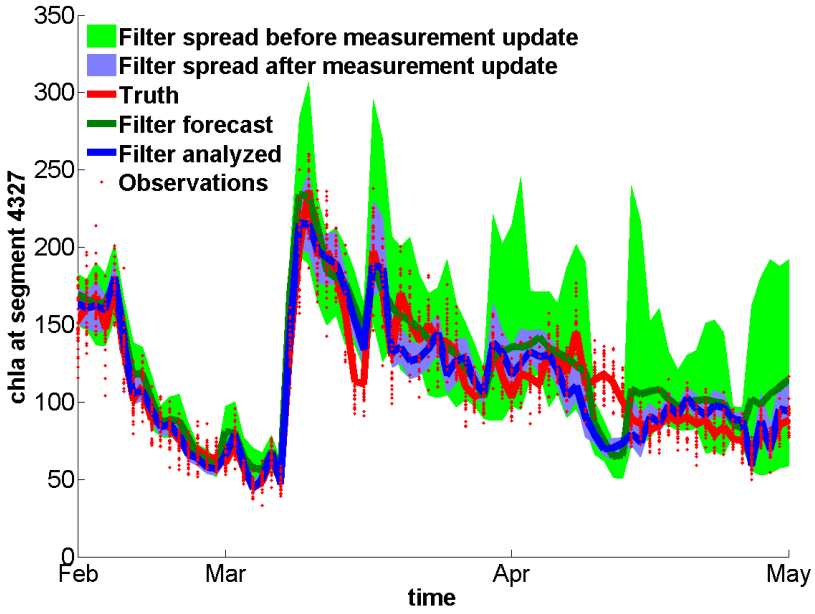


Figure 65: ***This figure shows states of the localized ESRF 2 before and after measurement update, of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

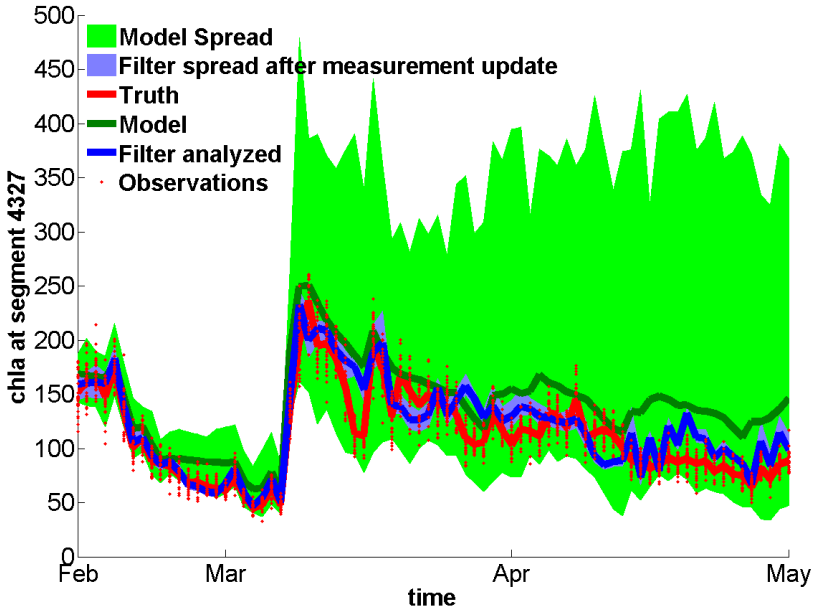


Figure 66: ***This figure shows the model state and localized ESRF 3 state of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***

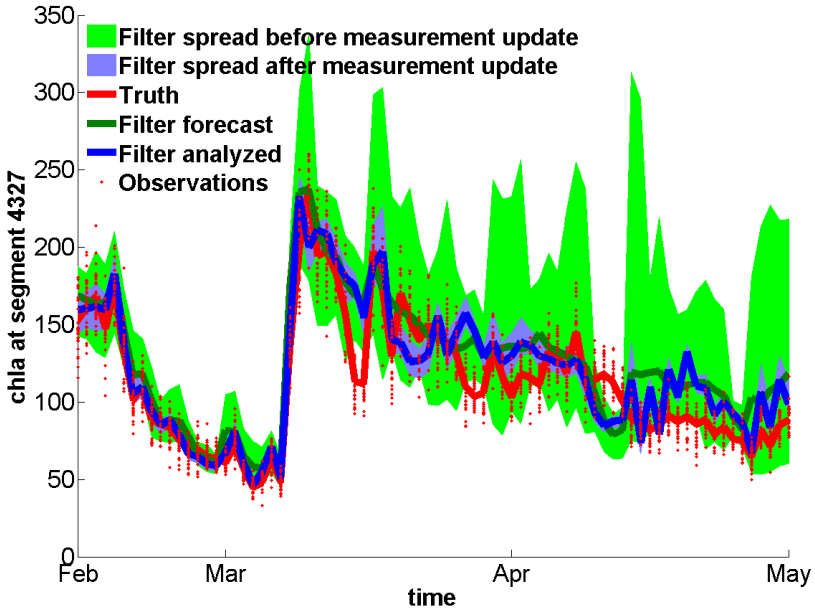
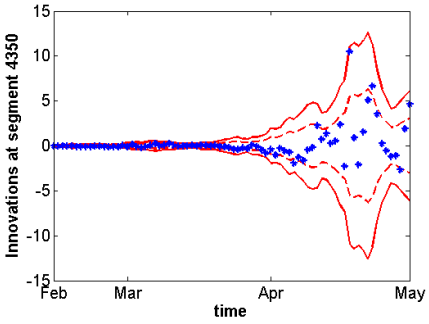
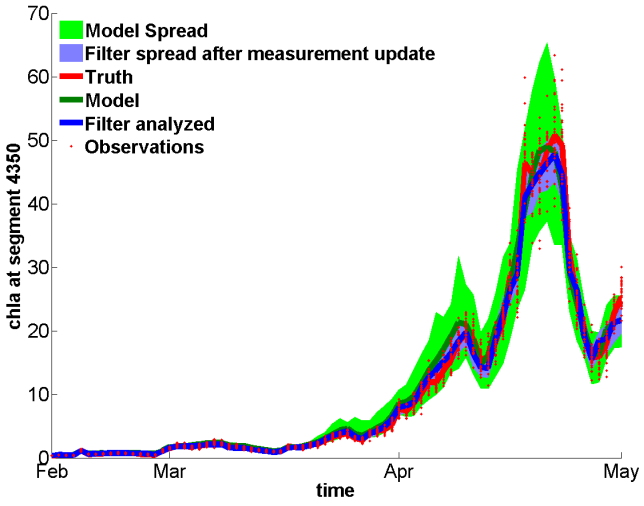


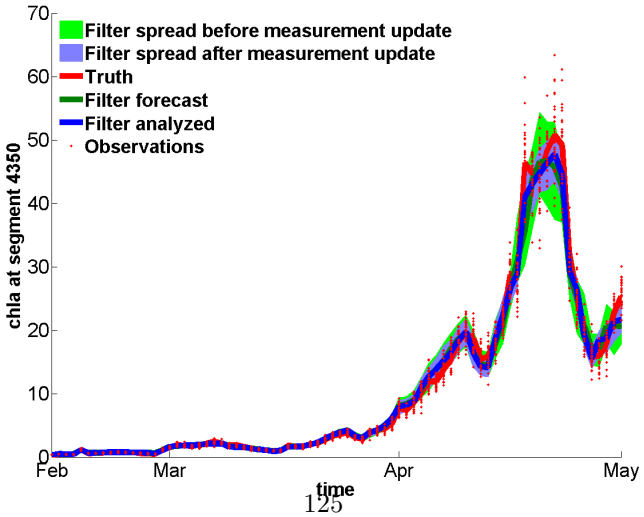
Figure 67: ***This figure shows states of the localized ESRF 3 before and after measurement update, of the Chlfa concentration at segment 4327. Also shown is the true state with observation noise.***



(a) Innovations of the EnKF. The dotted red line represents the σ bounds and the red line the 2σ bounds. The blue stars represent the estimated innovations. Percentages between bounds are [86,100]

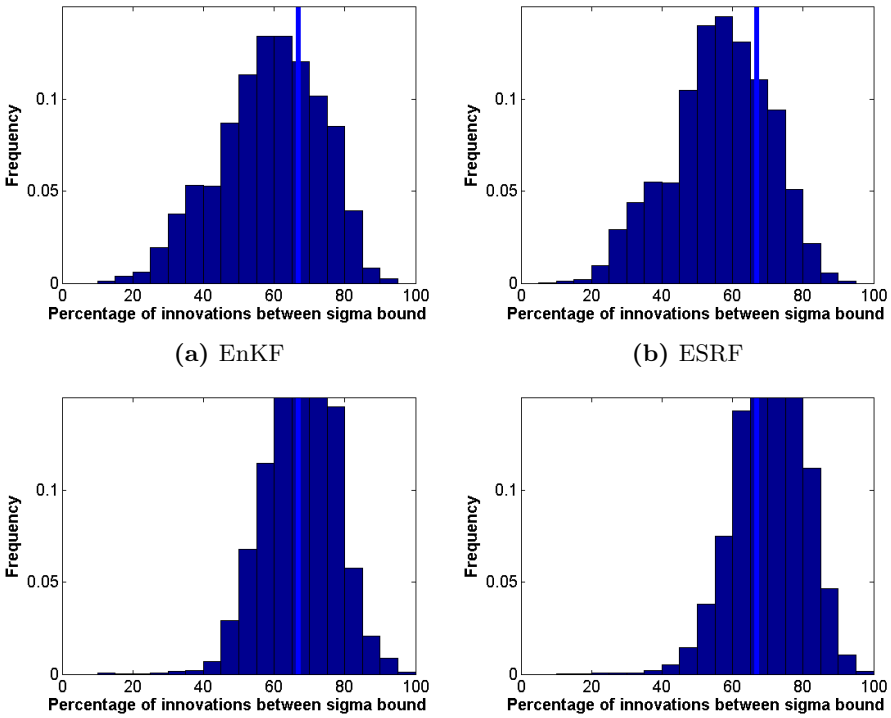


(b) Model state and EnKF state of the Chlfa concentration. Also shown is the true state with observation noise.



(c) States of the EnKF before and after measurement update, of the Chlfa concentration. Also shown is the true state with observation noise.

Figure 68: ***Example of a segment where ensemble collapse occurred with the EnKF***



(a) EnKF **(b)** ESRF

(c) ESRF localized 2 **(d)** ESRF localized 3

Figure 69: ***Frequency histograms of the percentages of innovations between sigma bounds, involved are the 4350 segments on the water surface. Indicated by a blue line is 67%, the value given by theory.***

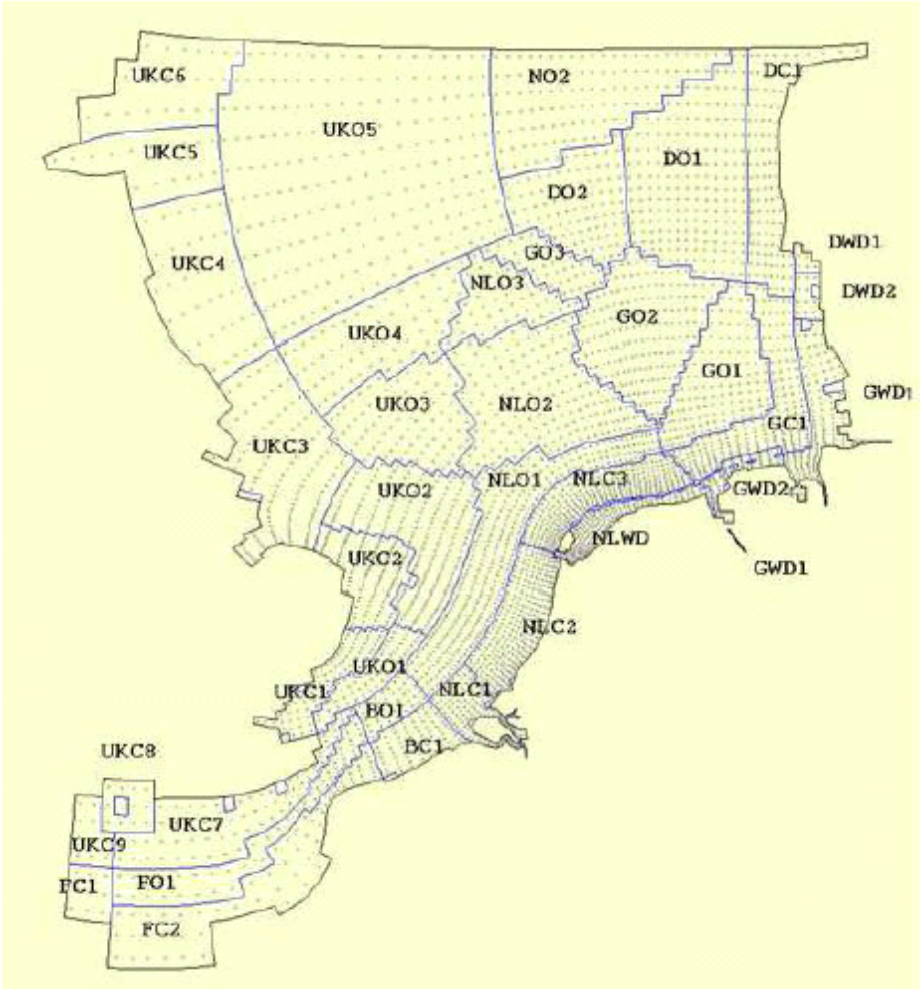


Figure 70: ***OSPAR regions***

16.2 List of figures

List of Figures

16.3 List of tables

List of Tables