PARAMETER, STATE AND UNCERTAINTY ESTIMATION FOR 3-DIMENSIONAL BIOLOGICAL OCEAN MODELS

by

Jann Paul Mattern

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
August 2012

DALHOUSIE UNIVERSITY

DEPARTMENT OF MATHEMATICS AND STATISTICS

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance a thesis entitled “PARAMETER, STATE AND UNCERTAINTY ESTIMATION FOR 3-DIMENSIONAL BIOLOGICAL OCEAN MODELS” by Jann Paul Mattern in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

Dated: August 15, 2012

External Examiner:

Research Co-Supervisors:

Examining Committee:

Departmental Representative:

DALHOUSIE UNIVERSITY

DATE: August 15, 2012

AUTHOR: Jann Paul Mattern

TITLE: PARAMETER, STATE AND UNCERTAINTY ESTIMATION FOR 3-DIMENSIONAL BIOLOGICAL OCEAN MODELS

DEPARTMENT OR SCHOOL: Department of Mathematics and Statistics

DEGREE: PhD CONVOCATION: October YEAR: 2012

Signature of Author

To my family and friends

Table of Contents

LIST OF TABLES

List of Figures

ABSTRACT

Realistic physical-biological ocean models pose challenges to statistical techniques due to their complexity, nonlinearity and high dimensionality. In this thesis, statistical data assimilation techniques for parameter and state estimation are adapted and applied to biological models. These methods rely on quantitative measures of agreement between models and observations. Eight such measures are compared and a suitable multiscale measure is selected for data assimilation. Build on this, two data assimilation approaches, a particle filter and a computationally efficient emulator approach are tested and contrasted. It is shown that both are suitable for state and parameter estimation. The emulator is also used to analyze sensitivity and uncertainty of a realistic biological model. Application of the statistical procedures yields insights into the model; e.g. time-dependent parameter estimates are obtained which are consistent with biological seasonal cycles and improves model predictions as evidenced by cross-validation experiments. Estimates of model sensitivity are high with respect to physical model inputs, e.g river runoff.

LIST OF ABBREVIATIONS AND SYMBOLS USED

Abbreviation	Description
ADA	Asynchronous Data Assimilation
AGB	Adapted Grey Block Distance
AHD	Adapted Hausdorff Distance
AVG	Averaged Distance
CDOM	Colored Dissolved Organic Matter
C:Chl	Carbon to Chlorophyll ratio
DG	Delta-G Distance
EnKF	Ensemble Kalman Filter
IE	Image Euclidean Distance
IR	Instant Remineralization
MAB	Middle Atlantic Bight
MABGOM	Middle Atlantic Bight and Gulf of Maine
MODIS	Moderate Resolution Imaging Spectroradiometer
NENA	Northeast North American
NPZD	Nutrient, Phytoplankton, Zooplankton, Detritus
NXC	Normalized Cross-Correlation
pdf	probability density function
RGB	Red, Green, Blue
RMS	Root Mean Square
RMSE	Root Mean Square Error
ROMS	Regional Ocean Modeling System
sd	standard deviation
SeaWiFS	Sea-viewing Wide Field-of-view Sensor
SIR	Sequential Importance Resampling
SOD	Sediment Oxygen Demand

Symbol	Description
$\#_{\textit{real}}$	function returning the number of non-*NaN* values in an image
ϵ_{trunc}	truncation error
ϵ_{quad}	quadrature error
θ_1	model parameter: maximum chlorophyll to carbon ratio
θ_2	model parameter: maximum grazing rate of zooplankton
w	parameter for the adapted grey block distance
a_{weight}	parameter in the particle filter weighting procedure
B	the beta function
$d^*_{\textit{NaN}}$	function for use with the adapted Hausdorff distance
D, M_1, M_2	images for use in Chapter 2
k_{max}	maximum order in polynomial chaos approximation
NaN	missing value
n_{ADA}	gap in ADA procedure
n_{ens}	number of ensemble members
p_{hypox}	probability of hypoxia

ACKNOWLEDGEMENTS

I want to thank my supervisors Katja Fennel and Mike Dowd for their support and encouragement. They let me go my own way for many of my projects, yet had always time to help. I also want to thank my committee members Keith Thompson and Bruce Smith for their helpful comments and discussions. Thanks also go my external examiner Peter Jan van Leeuwen for his constructive comments and questions regarding this thesis.

Many thanks also go to past and present members of my “lab”, the Marine Environmental Modelling Group. Special thanks to the ROMS experts Laura Bianucci and Arnaud Laurent who helped me tame the 3-dimensional biological ocean models that appear in the thesis title.

Thanks also to all my friend and fellow LSC inhabitants who made my time in the concrete bunker more social, enjoyable and sometimes more bizarre. There are too many to list them all, some of those I could talk science to over a drink are (in no particular order): Alan Hidy, Angela Kuhn, Bob Farmer, Catalina Gomez, Diego Ibarra, Jessica Carrière-Garwood, Jorge Urrego, Neil Davies, and Ramón Filgueira.

For their expert technical support I also want to thank Balagopal Pillai and Daniel Morrison.

Finally, I would like to thank ONR for funding my PhD research.

CHAPTER 1

INTRODUCTION

Over recent decades, 3-dimensional numerical models have become important tools in oceanography. Today, they are used for a variety of purposes, e.g. increasing our knowledge about oceanic processes, reconstructing the past, or forecasting the state of the ocean. Ocean models help to fill observation gaps, shed light on unknown mechanisms and provide evidence for, or against, scientific theories. With the constant rise in computing power, ocean models continue to grow in their capabilities and complexity.

The application of statistical procedures to ocean models presents a challenge due to the models' complexity. Ocean models are typically high-dimensional, nonlinear and require a long runtime, which severely limits the number of model evaluations that can be performed. Nevertheless, statistical techniques are needed to objectively assess and improve the abilities of ocean models, to gain a better understanding of the models, and to increase their value as scientific tools. This thesis presents the development and application of statistical techniques to biological ocean models. Existing techniques are expanded, and adapted, for the use with complex models, with the goal of examining, improving and comparing biological ocean models with observations.

Biological ocean models are a subclass of ocean models that are typically aimed at simulating biological processes at lower trophic levels, i.e. processes that relate to organisms such as phytoplankton which are at the bottom of the food web. As these biological processes are strongly connected to the chemistry and physics of the ocean, biological ocean models often contain a simple representations of chemical and physical processes. For this reason biological models are also referred to as *physical-biological*, or *biogeochemical* models. The close coupling of the lower trophic biology to physical and chemical processes in the ocean that makes them an important research area: as the bottom of the food web, phytoplankton are the basis of life in the ocean. Phytoplankton account for half of the earth's primary production (*Field et al.*, 1998). Biological processes at the lower trophic levels control the export of carbon from the atmosphere to the deep ocean, significantly affecting world climate (*Field et al.*, 1998). A better understanding of the lower trophic level processes thus helps to explain and predict events of global importance.

The first biological ocean model goes back to *Riley* (1946), who developed and analyzed a spatially-averaged model of phytoplankton development on Georges Bank. This model contained just one variable, phytoplankton, and was simple enough to be solved without the help of a computer. Since then, biological ocean models have become increasingly complex and require significant computing power. A typical biological ocean model of today simulates the carbon cycle in the upper ocean and contains variables for phytoplankton and zooplankton, detritus and nutrients. If they are of interest, other biological or chemical properties such as dissolved oxygen are included in the model as well. All variables are embedded in a 3-dimensional physical submodel which simulates their physical environment including water circulation and mixing.

The famous quote "Essentially, all models are wrong, but some are useful." (*Box and Draper*, 1987) holds true for biological ocean models, especially with regard to the complex processes that occur in the ocean. As biological ocean models are becoming more commonly used scientific tools, there is a growing need to develop methods to examine their usefulness and improve them objectively. The fundamental part of assessing the capabilities of models is the comparison with observational data, or real world measurements. The definition of methods and metrics to quantitatively measure the agreement between observations and model predictions is an essential part of model assessment.

For biological ocean models, satellite observations of surface chlorophyll provide an invaluable source of data. These observations are images of the ocean taken by satellites depicting the chlorophyll concentration close to the surface. The *Coastal Zone Color Scanner*, launched in 1978 was the first dedicated satellite sensor to provide ocean colour information which is used to derive chlorophyll concentrations (*Mitchell*, 1994). Today, satellite images of chlorophyll are widely available at a high temporal and spatial resolution, cover a large area, and thus have a great potential for model-data comparison. However, missing values and high levels of observation error, especially close to the coast, are characteristics of chlorophyll satellite data which make their use more difficult. In Chapter 2, eight image comparison measures aimed especially at the comparison of biological ocean models with satellite images of chlorophyll, are presented and compared. The image comparison measures that are examined, range from simple techniques such as the root mean square error and image correlation (both adapted to handle missing values) to more complex procedures taken and adapted from the computer vision literature. In a series of rigorous tests, the measures' performance in typical scenarios, including increasing levels of missing values and observation error, is evaluated. One measure, the *adapted grey block distance* that compares model and data at multiple resolutions, is shown to perform well throughout all of the tests. It forms the basis of the data assimilation studies in Chapter 3 and 4.

One step up from model-data comparison is the improvement of models using observations. For this purpose, a variety of statistical procedures has been developed, which assimilate, or incorporate, observations into numerical models. These procedures are collectively known as *data assimilation* procedures (*Kitagawa*, 1996; *Doucet et al.*, 2000). Data assimilation can be divided into two broad categories, state and parameter estimation, both of which are important in the context of biological ocean models. State estimation aims to improve the model state, one or more of the model's variables, without changing the parameters of the model. They are typically sequential procedures which use a series of observations to assimilate them in sequence at the time the observations were made. The aim of state estimation is generally to increase model accuracy for a time period in the past with available data, or improve the model's short term forecasting skill by improving initial conditions. Parameter estimation procedures use observations to improve the agreement between model and observations by adjusting the parameters of the model. The goals of parameter estimation are manyfold, as improved parameters can be used to obtain better state estimates and to learn about the processes that are modelled via the parameters. Parameter estimation plays an important role in biological ocean models because biological parameters are often poorly known.

The simplified representation of plankton species in typical biological models motivates the parameter estimation study in Chapter 3. While biological models have been growing in complexity, they often represent phytoplankton and zooplankton by one or just a few variables. This implies that a variety of different plankton species are lumped together into single variables, and thus each variable represents an inhomogeneous group of organisms. The parameters associated with plankton variables therefore are a simplified version of reality and describe an average property of the species composition represented by each variable. As phytoplankton populations experience strong seasonal shifts and spatial variation, there is little reason to assume that species composition in the variables stays fixed in time or space. To examine if changes in plankton compositions are reflected in their parameters, time- and space-dependent parameter estimation experiments are performed in Chapter 3, based on a year-long series of daily chlorophyll satellite observations and the comparison measure introduced in Chapter 2.

Since the experiments in Chapter 3 require many model evaluations, an emulator approach that saves computing time is used for parameter estimation. Emulators are model surrogates that approximate model output based on an existing set of model simulations. The emulator approximations can be obtained considerably faster than regular model estimation (within seconds as opposed to days) and thus allows a large number of experiments needed to obtain parameter estimates at a fine temporal resolution. Using the emulator, two representative plankton parameters are examined for both temporal and spatial dependence. The development of the parameters is assessed using a cross-validation procedure and contrasted with expected development based on biological processes.

Based on the results in the previous chapter, Chapter 4 examines the capabilities of a particle filter using the same environment, model and observations as the emulator approach. Particle filters are statistical procedures for data assimilation that can be used for both state and parameter estimation. They require no assumptions about the underlying model and are thus very versatile and suitable for highly nonlinear applications. In the realm of particle filters, the model state is assumed to be uncertain and to have a probability distribution. Particle filters approximate this distribution with an ensemble of model simulations which represent a sample from that distribution. Despite their versatility, the application of particle filters has been impeded by an inadequate representation of the probability distribution by the ensemble in many high-dimensional applications (*van Leeuwen*, 2009). For this reason, the use of particle filters for ocean models is not widespread. The implementation of a particle filter in Chapter 4 includes modifications to achieve a suitable ensemble representation while keeping the ensemble size low to satisfy computational constraints. Emulator results from Chapter 3 provide a reference solution that helps to evaluate the particle filter's performance in this high-dimensional problem and its general applicability to biological ocean models.

While Chapters 2 to 4 focus on observation-driven procedures to analyze and improve models, Chapter 5 contains a thorough examination of a typical biological ocean model without the use of data. Here, a combined model sensitivity and uncertainty analysis is performed to analyze how uncertainty in the model's inputs (e.g. parameters, initial and boundary conditions) is propagated to its output. Model sensitivity and model uncertainty are two properties that are inherently connected. Model sensitivity describes the response of the model output to changes in its inputs; a model is said to be sensitive to a particular input if a small change in the input causes a large response in the output. Since many inputs to ocean model contain uncertainty, the model's sensitivity determines how that uncertainty is expressed in the model's output. Knowledge about a models sensitivity therefore allows for statements regarding the spatial and temporal distribution of uncertainty in the model. It also determines how a reduction in model uncertainty can best be achieved, as reducing the error in a sensitive input is an effective way to diminish model uncertainty.

An uncertainty analysis involves sampling the model inputs and performing model simulations for each sample. In light of the high computational cost for each simulation, the uncertainty analysis in Chapter 5 utilizes the model emulator from Chapter 3 to approximate model output and reduce the number model evaluations. The aim in Chapter 5 is to identify significant sources of uncertainty in the biological output of a typical model, examine temporal and spatial patterns in model uncertainty, and to evaluate the emulator's capabilities for performing such a study.

Together, the following four chapters present an analysis of different aspects of typical biological ocean models. Suitable methods for dynamical numerical model analysis are identified and adapted and the new methods are used to gain general insights into biological ocean models.

CHAPTER 2

INTRODUCTION AND ASSESSMENT OF MEASURES FOR QUANTITATIVE MODEL-DATA COMPARISON USING SATELLITE IMAGES[1]

2.1 Introduction

In applications that involve satellite data and numerical modelling, it is desirable to compare the model output to measured data quantitatively. In this chapter, I assess the use of algorithms from the field of computer vision that measure the similarity of two images (henceforth referred to as image comparison measures) for model-data comparison. The image comparison measures proposed in this chapter offer an unexplored alternative to the measures currently used in model skill assessment and data assimilation.

Model skill assessment relies on similarity measures that quantify the distance of data and model output (see e.g. *Allen et al.*, 2007; *Stow et al.*, 2009, for model skill assessment in oceanographic contexts). As a quick and easy measure, the root mean square error is frequently used (see e.g. *Lehmann et al.*, 2009). In variational data assimilation, a cost function is defined measuring the discrepancy between data and their corresponding model counterparts (*Bennett*, 2002), usually a mean square error. In ensemble-based data assimilation, which includes particle filters, a likelihood function must be defined, specifying the observation or model errors (*Dowd*, 2007).

Satellite data typically come as single-channel (as opposed to multi-channel images,

[1] based on: Mattern, J. P., K. Fennel, and M. Dowd, Introduction and Assessment of Measures for Quantitative Model-Data Comparison Using Satellite Images, *Remote Sensing, 2*, 794–818, 2010b.

such as RGB images, which contain 3 channels, one for each red, green and blue colour information) digital images, *i.e.*, numeric values arranged on a grid. Hence I can consider model-data comparison to be the comparison of two single-channel digital images: one representing the data, the other the model state. I focus specifically on the comparison of ocean colour data to corresponding data derived from numerical oceanographic models.

In the context of computer vision, a variety of image comparison methods have been developed for the comparison of regular (single-channel, discrete-valued) grey scale images (see e.g. *Avcıbaş et al.*, 2002, and references therein). Satellite ocean colour images exhibit two major differences from regular images: (1) intensity values are generally not discrete, and (2) satellite images often contain regions of missing values caused by cloud cover or other atmospheric distortions (*Mannino et al.*, 2008), as well as masked regions due to the presence of land (islands, coastline). Missing values especially pose a challenge to existing image comparison measures and here I present suitable adaptations. Although I focus on ocean colour, the methods can be readily applied to other variables such as sea surface temperature and sea surface height. I also broaden the definition of an image. While regular grey scale images contain discrete values arranged on a complete grid, I allow non-discrete and missing values. Because of the similarity to regular images, I use the words *image* and *pixel* even though I am not dealing with regular images.

Image comparison measures can be divided roughly into two categories, both are widely used in different applications (*Avcıbaş et al.*, 2002; *Lehmann et al.*, 1997; *Le Moigne and Tilton*, 1995; *Alberga*, 2009). One category, high level image comparison, incorporates edge detection (see *Holyer and Peckinpaugh*, 1989; *Cayula and Cornillon*, 1992; *Belkin and O'Reilly*, 2009, for oceanographic examples), or other segmentation methods to extract features from images (*Nichol*, 1987). Another approach in high level image comparison is to find a mapping to morph one image into another (see *Beezley and Mandel*, 2008, for a use in data assimilation). The extracted features or morphed images are then classified or compared in place of the images. Low level image comparison, the second category, consists of direct comparison of images as a whole. In this chapter, I focus on various approaches of low level image comparison and hereafter the term image comparison refers to this category only.

Some of the simplest low level image comparison methods utilize pixel-by-pixel comparison, *i.e.*, only pixels at the same location are compared (e.g., root mean square error and correlation). In the computer vision literature it is often pointed out that pixel-by-pixel comparison, while having certain advantages (mainly simplicity and low run-time), often reflects human perception poorly and are too sensitive to small changes within an image (*Santini and Jain*, 1999). A small offset of one or multiple objects within an image can, for example, cause the root mean square error to increase dramatically. For this reason it is desirable not to restrict the comparison to pixels at the same location, but to include neighbourhoods of pixels.

In this chapter I compare the performance of 8 image comparison measures after adapting them to allow for missing and non-discrete values. Among the tested methods, the root mean square error and the normalized cross-correlation represent widely used pixel-by-pixel measures (see e.g. *Avcıbaş et al.*, 2002; *Di Gesú and Starovoitov*, 1999). I demonstrate that these pixel-by-pixel measures have shortcomings in the context of satellite image comparison. The results indicate the benefits of alternative, neighbourhood-based methods. The shortcomings of root mean square error and cross-correlation become especially apparent with respect to missing values in images.

In this chapter I use ocean colour images from the SeaWiFS and MODIS satellites. The raw images were processed with the algorithm of *Mannino et al.* (2008) to produce images that measure the absorption of light due to the organic constituent coloured dissolved organic matter (CDOM) and chlorophyll.

This chapter is organized as follows. In Section 2.2 I define the nomenclature. Section 2.3 contains descriptions of the 8 image comparison measures to be tested. I evaluate the performance of the image comparison measures in a series of tests in Section 2.4. The results are summarized in Section 2.5.

2.2 Nomenclature

2.2.1 Definition of Symbols

I consider a digital image A as a set of pixels

$$A = \{a_{i,j}\}_{i,j=1}^{m,n} \tag{2.1}$$

with values $a_{i,j} \in [0, g] \cup \{\mathit{NaN}\}$. The symbol *NaN* indicates a missing value in a pixel, $[0, g]$ is the closed interval from 0 to g and $\cup$ denotes the union symbol for two sets. A is defined on a $m \times n$ grid

$$X = \{(i,j)\}_{i,j=1}^{m,n} \tag{2.2}$$

so that the pixel $a_{i,j}$ is located at $(i,j) \in X$.

In the following I use $A = \{a_{i,j}\}_{i,j=1}^{m,n}$ and $B = \{b_{i,j}\}_{i,j=1}^{m,n}$ to denote model and satellite images, respectively. I assume that both images are defined on the same grid X. Further, I define A_{NaN} and A_{real} as the set of all points in X where the pixels of A are *NaN* (missing value) and not *NaN* (real valued) respectively. Together they form a partition of X

$$A_{\mathit{NaN}} = \{(i,j) \in X : a_{i,j} = \mathit{NaN}\} \qquad \text{and} \qquad A_{\mathit{real}} = X \setminus A_{\mathit{NaN}}, \tag{2.3}$$

where $\setminus$ denotes the relative complement of two sets. In the same way, I also define a partition of B:

$$B_{\mathit{NaN}} = \{(i,j) \in X : b_{i,j} = \mathit{NaN}\} \qquad \text{and} \qquad B_{\mathit{real}} = X \setminus B_{\mathit{NaN}}. \tag{2.4}$$

Missing values in model and data images present a challenge to similarity measures because they cannot be compared to other values in a meaningful way. Generally, the location of missing values in A will not correspond to the location of missing values in B, and *vice versa*, so that $A_{\mathit{NaN}} \neq B_{\mathit{NaN}}$. This leads to the formation of 4 subregions on the grid which form a partition of X and need to be treated differently by the image similarity measures. These subregions are:

- $A_{\mathit{NaN}} \cap B_{\mathit{NaN}}$,
- $A_{\mathit{real}} \cap B_{\mathit{real}} = X_{\mathit{real}}$,
- $A_{\mathit{real}} \cap B_{\mathit{NaN}}$ and
- $A_{\mathit{NaN}} \cap B_{\mathit{real}}$,

where $\cap$ denotes the intersection of two sets.

The region of pixels with missing values in both model and data, $A_{\mathit{NaN}} \cap B_{\mathit{NaN}}$, is ignored by all similarity measures presented here. These pixels therefore should not influence a given similarity measure's result. The size of $A_{\mathit{NaN}} \cap B_{\mathit{NaN}}$ relative to the size of X may, however, affect the confidence in the similarity measures result. In this application $A_{\mathit{NaN}} \cap B_{\mathit{NaN}}$ consists solely of land pixels.

Pixels within the subregion of X with non-*NaN* values in both model or data, $X_{real} = A_{real} \cap B_{real}$, can be treated like those in regular images. Pixel-by-pixel comparison measures, which do not consider the distance between two pixels at different locations, such as the root mean square error, base their results solely on this region. Pixels with missing values either in A or B ($A_{NaN} \cup B_{NaN}$) are not taken into account by these pixel-by-pixel measures, while other similarity measures presented herein can make use of them.

Cloud cover and a variety of different distortions can lead to missing values in satellite images. Additionally, the satellite image may not cover the entire model domain. Pixel locations in $A_{real} \cap B_{NaN}$, with values in the model but missing values in the data are therefore common for satellite data. They can become important for those similarity measures that compare the distance between two pixels at different locations.

Pixel locations with values in the data but not in the model, in $A_{NaN} \cap B_{real}$, are not considered in this chapter, as I do not make use of data that extends beyond the model domain. Generally it is possible to consider this region e.g., when data from outside the model domain is available. In that case the neighbourhood-based measures introduced here can make use of the additional data. The 3 remaining regions considered here are illustrated in Figure 2.1.

2.3 Image Comparison Methods

Most of the image comparison measures that I assess here require modifications to work with missing values (Table 2.1). Dealing with missing values in pixel-by-pixel measures is trivial, as pixels with missing values in A or B can only be ignored by these measures. I consider the widely used root mean square error and normalized cross-correlation, and, as a less widely used pixel-by-pixel distance measure, I include an adaptation of the entropic distance D_2 (***Di Gesú and Roy***, 2002) in my tests. The remaining 5 distance measures presented here are based on the comparison of neighbouring pixels and require modifications to make use of the information in $A_{real} \cap B_{NaN}$, where only one of the two images have non-missing values. I altered the image Euclidean distance and Delta-g (Δ_g) only slightly from their original formulations while I changed the adapted Hausdorff distance, the averaged distance and the adapted grey block distance significantly.

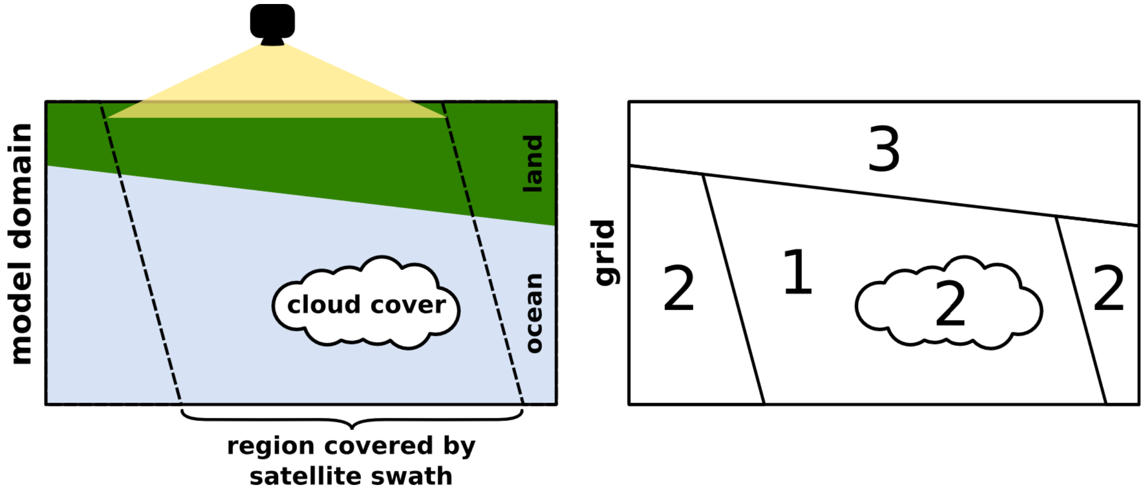


Figure 2.1: Schematic of a satellite taking an image (left panel) and corresponding subregions on the model grid (right panel). Subregion 1 in the right panel corresponds to $X_{real} = A_{real} \cap B_{real}$ for the satellite image A and model B, where the view of the ocean is clear. The satellite image does not cover the entire model domain and clouds as well as other interferences cause missing values in the data, creating subregion 2 ($A_{real} \cap B_{NaN}$). The model domain includes land, resulting in subregion 3 ($A_{NaN} \cap B_{NaN}$).
.

2.3.1 Parametrisations

Most of the 8 image comparison measures I introduce in the following sections feature one or more parameters. Table 2.1 lists the parametrizations that I used here and identifies the scale parameters for the neighbourhood-based measures.

In the following, I refer to an image comparison measure (in its original formulation or my adaptation) by its full name, e.g., “image Euclidean distance” while I use its abbreviation, e.g., “IE”, to refer to the particular parametrization of the measure that I used in my tests. Table 2.1 contains a list of the abbreviations; in case of the adapted Hausdorff distance, I test two different parametrizations, AHD and AHD2.

The choice of parameters used with the methodologies, including my choice of using the entropic distance D_2 over e.g., D_1 (both were introduced by *Di Gesú and Roy* (2002)), are based on results I obtained from an initial set of tests performed for a range of likely parameters, as well as on parameter values suggested in the original literature.

2.3.2 Pixel-by-Pixel Measures

As stated in Section 2.2, all pixel-by-pixel measures listed below simply ignore model pixels that correspond to data pixels with missing values. They operate only on X_{real}, the region where neither A or B have missing values.

image comparison measure	p-b-p[a]	reference	scaling param.	name	parametrization	complexity $(O(\bullet))$
root mean square error	✓			RMS		nm
normalized cross-correlation	✓			NXC		nm
entropic distance D_2	✓	*Di Gesú and Roy* (2002)		D2		nm
adapted Hausdorff distance		*Huttenlocher et al.* (1993)	g_{max} [c]	AHD[d]	$p \to \infty$, $k_{\text{max}} = 1$	$(mn)^2$
				AHD2[d]	$p \to \infty$, $k_{\text{max}} = 20$	
averaged distance		*Di Gesú and Starovoitov* (1999)	w, g_{max} [c]	AVG [d]	$w = 10$	$(m-w)\times(n-w)\times(2w)^2$
Delta-g (Δ_g)		*Wilson et al.* (1997)	c [b]	DG	$n_{\text{lev}} = 32$, $c = 20$	$n_{\text{lev}}(mn)^2$
image Euclidean distance		*Wang et al.* (2005)	σ	IE	$\sigma = 1$	$(mn)^2$
adapted grey block distance		*Juffs et al.* (1998)	w	AGB	$w = 1$	$\log_2(\max(m,n))\times\max(m,n)^2$ [e]

[a] pixel-by-pixel
[b] see *Wilson et al.* (1997)
[c] relative to m_{max}, n_{max}
[d] AHD, AHD2, AVG use $g_{\text{max}} = g$, $m_{\text{max}} = n_{\text{max}} = \max(m, n)$
[e] see Section 2.3.3 for more details

Table 2.1: The comparison measures and the names of their parametrizations used in the tests in Section 2.3.

2.3.2.1 Root Mean Square Error (RMS)

The Root Mean Square (RMS) error is defined as

$$d_{\mathrm{RMS}}\left(A,B\right)=\sqrt{\frac{1}{|X_{real}|}\sum_{(i,j)\in X_{real}}\left(a_{i,j}-b_{i,j}\right)^{2}}, \tag{2.5}$$

where $|X_{real}|$ is the number of pixels in X_{real}.

2.3.2.2 Normalized Cross-Correlation (NXC)

The Normalized Cross-Correlation ratio (NXC, *Di Gesú and Starovoitov* (1999)), is a pixel-by-pixel comparison measure based on cross-correlation that is closely related to the root mean square error and also widely used. It has the form

$$d_{\mathrm{NXC}}\left(A,B\right)=1-\frac{\sum_{(i,j)\in X_{real}}a_{i,j}b_{i,j}}{\sqrt{\sum_{(i,j)\in X_{real}}a_{i,j}^{2}\sum_{(i,j)\in X_{real}}b_{i,j}^{2}}}. \tag{2.6}$$

2.3.2.3 Entropic Distance D_2 (D2)

The Entropic Distance D_2 (D2) is one of 6 entropy-based image comparison measures introduced by *Di Gesú and Roy* (2002). As with RMS, pixels with missing value are ignored by D2. The distance between two images A and B is then defined as

$$d_{\mathrm{D_2}}\left(A,B\right)=\frac{1}{|X_{real}|}\sum_{(i,j)\in X_{real}}2\ h_{i,j}(A,B)\ \left(1-\frac{h_{i,j}(A,B)}{2}\right) \tag{2.7}$$

where $h_{i,j}(A,B)=\frac{|a_{i,j}-b_{i,j}|}{g}$ and g is the highest intensity value in the images (see Section 2.2), ensuring $\frac{|a_{i,j}-b_{i,j}|}{g}\in[0,1]$. This measure is based on an extended definition of Shannon entropy (*Shannon*, 2001), it is a true metric (*Di Gesú and Roy*, 2002). The factor 2 in the numerator in the above equation ensures that each term $2\ h_{i,j}(A,B)\ \left(1-\frac{h_{i,j}(A,B)}{2}\right)$ is monotonically increasing in $h_{i,j}$ and does not reach its maximum at $\frac{1}{2}$ which is true for the Shannon entropy (see Section 5.2.4).

2.3.3 Neighbourhood-Based Measures

Neighbourhood-based measures utilize a variety of methods that allow for the comparison of non-neighbouring pixels. With the exception of the adapted grey block distance, all of the comparison measures in this section make use of a direct approach to pixel comparison by defining a measure for the distance between pixels. A common definition for the distance between two non-*NaN* pixels $a_{i,j}$ and $b_{k,l}$ is

$$d^*(a_{i,j}, b_{k,l}) = \left(\frac{|a_{i,j} - b_{k,l}|^p}{g_{\text{max}}} + \frac{|i-k|^p}{m_{\text{max}}} + \frac{|j-l|^p}{n_{\text{max}}} \right)^{1/p}, \tag{2.8}$$

which measures the distance in the 3-dimensional space spanned by intensity, horizontal and vertical pixel location. For $p = 1$, d^* is called city block distance, for $p = 2$, d^* corresponds to the Euclidean distance and for $p \to \infty$ to the maximum distance

$$d^*_{\text{max}}(a_{i,j}, b_{k,l}) = \max \left(\frac{|a_{i,j} - b_{k,l}|}{g_{\text{max}}}, \frac{|i-k|}{m_{\text{max}}}, \frac{|j-l|}{n_{\text{max}}} \right). \tag{2.9}$$

The constants g_{max}, m_{max} and n_{max} in Equations (2.8) and (2.9) are scaling parameters. As suggested by *Di Gesú and Starovoitov* (1999), I set them to $g_{\text{max}} = g$ and $m_{\text{max}} = n_{\text{max}} = \max(m, n)$, where g is the maximum intensity value in the images and m and n are the dimensions of A and B. The pixel distance d^* is used in the original formulation of the Hausdorff distance, the averaged distance and Delta-g. To account for *NaN*-valued pixels, I extend d^* in the following sections and discuss it in more detail.

2.3.3.1 Adapted Hausdorff Distance (AHD, AHD2)

The Hausdorff distance was presented by *Huttenlocher et al.* (1993) and has been used in numerous variations (e.g. *Dubuisson and Jain*, 1994; *Sim et al.*, 1999). I present two adaptations of the Hausdorff distance (AHD, AHD2) based on its original and most common definition (*Huttenlocher et al.*, 1993; *Di Gesú and Starovoitov*, 1999):

$$d_{\text{AHD}}(A, B) = \max_{i,j} \left(\max \left(d(a_{i,j}, B), d(b_{i,j}, A) \right) \right), \tag{2.10}$$

where $d(a_{i,j}, B)$ is a function defining the distance between a pixel in A and the entire image B (or a pixel in B and the entire image A for $d(b_{i,j}, A)$). This distance is commonly defined as

$$d(a_{i,j}, B) = \min_{k,l} \left(d^*_{\mathit{NaN}}(a_{i,j}, b_{k,l}) \right) \quad \text{and} \quad d(b_{i,j}, A) = \min_{k,l} \left(d^*_{\mathit{NaN}}(b_{i,j}, a_{k,l}) \right) \tag{2.11}$$

and includes the function d^*_{NaN} which I adapted for use with missing values: In addition to the 2 pixels $a_{i,j}$ and $b_{k,l}$, $d^*_{NaN}\left(a_{i,j}, b_{k,l}\right)$ is also dependent on $b_{i,j}$ in the following way

$$d^*_{NaN}\left(a_{i,j}, b_{k,l}\right) = \begin{cases} 0 & \text{if } a_{i,j} = \mathit{NaN} \text{ or } b_{i,j} = \mathit{NaN} \\ \infty & \text{if } b_{k,l} = \mathit{NaN} \text{ and } i \neq k \text{ or } j \neq l \\ d^*(a_{i,j}, b_{k,l}) & \text{otherwise,} \end{cases} \tag{2.12}$$

where d^* is the distance of two non-*NaN* pixels defined in equation (2.8). If $b_{i,j} = \mathit{NaN}$, this will result in $d^*_{NaN}\left(a_{i,j}, b_{k,l}\right) = 0$ independent of k and l. This property of d^*_{NaN} ensures that $d_{\text{AHD}}\left(A, B\right) = 0$ for $X_{real} = \varnothing$. Note that $d^*_{NaN}\left(a_{i,j}, b_{k,l}\right) \neq d^*_{NaN}\left(b_{k,l}, a_{i,j}\right)$; the symmetry in equation (2.10) ensures that d_{AHD} is symmetrical (*i.e.*, $d_{\text{AHD}}\left(A, B\right) = d_{\text{AHD}}\left(B, A\right)$). Furthermore, the adapted Hausdorff distance is equal to the original Hausdorff distance (*Huttenlocher et al.*, 1993) if there are no missing values in A and B.

The above definition of the adapted Hausdorff distance is sensitive to factors such as noise and missing values. To decrease this sensitivity, I average over the k_{max} largest values of $\max\left(d(a_{i,j}, B), d(b_{i,j}, A)\right)$, instead of just using the maximum as in equation (2.10). By defining $d_{\text{max}}(k)$ as the kth largest value of $\max\left(d(a_{i,j}, B), d(b_{i,j}, A)\right)$ for all $(i, j) \in X$, I can express the averaging as

$$d^{(k_{\text{max}})}_{\text{AHD}}\left(A, B\right) = \frac{1}{k_{\text{max}}} \sum_{k=1}^{k_{\text{max}}} d_{\text{max}}(k), \tag{2.13}$$

which is equivalent to the formulation in Equation (2.10) for $k_{\text{max}} = 1$, *i.e.*, $d^{(1)}_{\text{AHD}}\left(A, B\right) = d_{\text{AHD}}\left(A, B\right)$. In my tests I use two parametrizations of the adapted Hausdorff distance which only differ in their choice of k_{max}: for AHD $k_{max} = 1$, while $k_{max} = 20$ for AHD2. The averaging of the largest values described above has been used in *Sim et al.* (1999) to decrease the sensitivity of the Hausdorff distance to outliers. *Huttenlocher et al.* (1993) explored a similar idea for the Hausdorff distance and portions of regular images.

2.3.3.2 Averaged Distance (AVG)

The Averaged Distance (AVG) is an image comparison measure introduced by *Di Gesú and Starovoitov* (1999). I modified it to work with missing values as I have done for the adapted Hausdorff distance. The averaged distance is defined as the square root of averaged distances of sub-images of A and B:

$$d_{\text{AVG}}(A, B) = \sqrt{\frac{1}{\sqrt{2}n_{real}} \sum_{i=w}^{m-w} \sum_{j=w}^{n-w} \sqrt{\left(d(a_{i,j}, B_{w_{i,j}})\right)^2 + \left(d(b_{i,j}, A_{w_{i,j}})\right)^2}}, \tag{2.14}$$

where $A_{w_{i,j}}$ and $B_{w_{i,j}}$ are $w \times w$ sub-images of A and B, centered on $(i, j) \in X$. The distance d is defined in equation (2.11). The normalizing factor $n_{real} < (m - w)(n - w)$ is the number of summands in equation (2.14) for which $a_{i,j} \neq \mathit{NaN}$ and $b_{i,j} \neq \mathit{NaN}$. It is defined as

$$n_{real} = \left|\{(i, j) \in X : w \leq i \leq m - w,\ w \leq j \leq n - w,\ a_{i,j} \neq \mathit{NaN}, b_{i,j} \neq \mathit{NaN}\}\right|. \tag{2.15}$$

2.3.3.3 Delta-g (DG)

The distance measure Delta-g (DG) was introduced by *Wilson et al.* (1997) for grey scale image comparison. Due to its relatively complex definition, I will only give a simplified description of it here and explain the changes I made to the original formulation. In Delta-g, a grey scale image is viewed as a function of intensity y defined on a grid, *i.e.*, $A(i, j) = a_{i,j}$ for $(i, j) \in X$. For each intensity level y, I consider those pixels in A that have the same intensity level or are above it:

$$X(A, y) = \{(i, j) \in X : a_{i,j} \geq y\}. \tag{2.16}$$

This serves as a way to define a distance

$$d_{\text{DG}}((i, j), X(A, y)) = \min_{(k,l) \in X(A,y)} d((i, j), (k, l)) \tag{2.17}$$

where $d((i, j), (k, l))$ is the distance of two points in X, e.g. the Euclidean distance $d((i, j), (k, l)) = \sqrt{(i - k)^2 + (j - l)^2}$. Equation (2.17) is the basis of Delta-g and $d((i, j), X(A, y))$ is computed for each intensity level y. Here, Delta-g makes use of the discrete value characteristic of typical grey scale images, as there is only a finite number of intensity levels in a grey scale image (typically $y \in \{0, 1, 2, \ldots, 255\}$). I emulate this characteristic by mapping the intensity values of the satellite images from $[0, g]$ to their closest value on an equidistant grid $Y = \{0, \frac{g}{n_{\text{lev}}-1}, \frac{2g}{n_{\text{lev}}-1}, \ldots, g\}$, where n_{lev} is the number of levels. Equation (2.17) is then evaluated at every intensity level $y \in Y$. A higher n_{lev} typically improves the results of Delta-g but increases runtime significantly. The comparison of A and B is performed by introducing

$$d^*_{\text{DG}}(A, (i,j), y) = \min\left(c, \max_{y' \in Y}\left(d_{\text{DG}}((i,j), X(A,y')), |y - y'|\right)\right) \tag{2.18}$$

for a constant c and computing

$$\sum_{(i,j)\in X}\sum_{y\in Y}|d^*_{\text{DG}}(A, (i,j), y) - d^*_{\text{DG}}(B, (i,j), y)| \tag{2.19}$$

which is followed by a normalization to correct for the number of pixels. To deal with missing values, I ignore pixels in $A_{NaN} \cap B_{NaN}$ and define $d_{\text{DG}}((i,j), X(A,y)) = \infty$ (infinity) if $a_{i,j}$ is *NaN*.

2.3.3.4 Image Euclidean Distance (IE)

The Image Euclidean Distance (IE) is an Euclidean distance of two images, that considers the spatial links between different pixels (*Wang et al.*, 2005). While the RMS is the Euclidean distance of two images, assuming that all (mn) dimensions of A and B are orthogonal, the image Euclidean distance takes into account the grid structure, on which the pixels are located. I simply ignore missing values in the computation of the image Euclidean distance and define

$$d_{\text{IE}}(A, B) = \frac{1}{|X_{real}|}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{n}\hat{d}(a_{i,j}, b_{i,j}, a_{k,l}, b_{k,l}) \tag{2.20}$$

where

$$\hat{d}(a_{i,j}, b_{i,j}, a_{k,l}, b_{k,l}) = \begin{cases} 0 & \text{if } \#_{NaN}\{a_{i,j}, b_{i,j}, a_{k,l}, b_{k,l}\} > 0 \\ (a_{i,j} - b_{i,j})g_{i,j,k,l}(a_{k,l} - b_{k,l}) & \text{otherwise} \end{cases} \tag{2.21}$$

is computed for all combinations of pixels in A and B. The symbol $\#_{NaN}$ denotes the number of missing values in a set. The spatial distance between the pixels is incorporated into the Gaussian function $g_{i,j,k,l}$ which is defined as

$$g_{i,j,k,l} = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{(i-j)^2 + (k-l)^2}{2\sigma^2}\right). \tag{2.22}$$

As each pixel in A is compared with each pixel in B the complexity of IE is $O((mn)^2)$. This complexity can be reduced by comparing each pixel to only those pixels that are close to it, as other pixels will add near-zero terms to the sum in Equation (2.21). If only those pixels in a $w \times w$ window around each pixel are considered in the comparison, the complexity of IE reduces to that of AVG.

2.3.3.5 Adapted Grey Block Distance (AGB)

I also adapted the grey block distance, presented by *Juffs et al.* (1998) for regular grey scale images, to deal with missing values and refer to this as Adapted Grey Block Distance (AGB). In this measure the distance of two images is determined by comparing the mean grey level of successively smaller subdivisions (grey blocks) of the images.

Calculation of the grey block distance between two images A and B involves dividing A and B into blocks and comparing their mean intensity levels. This is done for different resolution levels, *i.e.*, the blocks are successively decreased in size and a comparison is performed at every level. For regular grey scale images the blocks must cover the image completely at every resolution (*Juffs et al.*, 1998), and the blocks cannot extend beyond the boundaries of the image. In my adapted grey block distance, d_{AGB}, I weight the difference between the mean intensity level of two blocks based on the number of missing values they contain. For two images A and B, the blocks must completely include the region where either A or B has non-*NaN* pixels at every resolution. However, A and B may be padded with missing values or embedded into larger images filled with missing values, effectively allowing blocks to extend beyond the boundaries of the original image.

To facilitate the division of A and B into successively smaller blocks, I embed both images into the centers of larger, *NaN*-filled, square images, A_{ext} and B_{ext}, respectively, with an edge length that is a power of two. For A and B of size $m \times n$, A_{ext} and B_{ext} are of size $2^{n_{\text{ext}}} \times 2^{n_{\text{ext}}}$ with $n_{\text{ext}} = \lceil \log_2 (\max(n, m)) \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling (round up to nearest integer) function. All other pixels of $A_{\text{ext}} = \{a_{i,j}^{(\text{ext})}\}_{i,j=1}^{2^{n_{\text{ext}}}}$ not defined by the embedding of A are missing values, so that

$$a_{i,j}^{(\text{ext})} = \begin{cases} a_{i-\delta_x, j-\delta_y} & \text{if } i \in \{\delta_x + 1, \dots, \delta_x + m\}, j \in \{\delta_y + 1, \dots, \delta_y + n\} \\ \textit{NaN} & \text{otherwise} \end{cases} \tag{2.23}$$

with $\delta_x = \left\lfloor \frac{1}{2} \left(2^{n_{\text{ext}}} - m\right) \right\rfloor$ and $\delta_y = \left\lfloor \frac{1}{2} \left(2^{n_{\text{ext}}} - n\right) \right\rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor (round down to nearest integer) function. The same applies to B_{ext}.

Beginning with the full image as a single square block, a series of increasingly smaller blocks is determined by dividing each block at the previous level into four equal quadrants. In this way, a $2^{n_{\text{ext}}} \times 2^{n_{\text{ext}}}$ image can be divided n_{ext} times until the block resolution is equal to the image's pixel resolution. For the r-th resolution level, the distance of A_{ext} and B_{ext} is defined as

$$d_{\text{AGB}}^{(r)}\left(A_{\text{ext}}, B_{\text{ext}}\right) = \frac{\sum_{j=1}^{2^{r-1}} \sum_{i=1}^{2^{r-1}} \left|\bar{a}_{i,j}^{(r)} - \bar{b}_{i,j}^{(r)}\right| \min\left(\#_{\mathit{real}}(\bar{a}_{i,j}^{(r)}), \#_{\mathit{real}}(\bar{b}_{i,j}^{(r)})\right)}{\sum_{j=1}^{2^{r-1}} \sum_{i=1}^{2^{r-1}} \min\left(\#_{\mathit{real}}(\bar{a}_{i,j}^{(r)}), \#_{\mathit{real}}(\bar{b}_{i,j}^{(r)})\right)} \tag{2.24}$$

which corresponds to a weighted L_1 norm. In the above equation $\bar{a}_{i,j}^{(r)}$ and $\bar{b}_{i,j}^{(r)}$ denote the mean intensity of the block at the coordinates i, j of A_{ext} and B_{ext}, respectively. The symbols $\#_{\mathit{real}}(\bar{a}_{i,j}^{(r)})$ and $\#_{\mathit{real}}(\bar{b}_{i,j}^{(r)})$ denote the number of non-*NaN* values in $\bar{a}_{i,j}^{(r)}$ and $\bar{b}_{i,j}^{(r)}$, respectively. Missing values are ignored in the calculation of the intensity mean for each block, if a block contains only missing values, its mean intensity is defined as 0. At the lowest resolution level ($r = 1$) one single block covers an entire image, at the highest level ($r = n_{\text{ext}} + 1$) each block contains a single pixel, so that

$$\bar{a}_{1,1}^{(1)} = \frac{1}{|A_{\mathit{real}}|} \sum_{(k,l)\in A_{\mathit{real}}} a_{k,l} \quad \text{and} \quad \bar{a}_{i,j}^{(n_{\text{ext}}+1)} = a_{i,j}^{(\text{ext})}. \tag{2.25}$$

The adapted grey block distance is then defined as a weighted sum of the distances at each resolution level:

$$d_{\text{AGB}}\left(A, B\right) = d_{\text{AGB}}\left(A_{\text{ext}}, B_{\text{ext}}\right) = \sum_{r=1}^{n_{\text{ext}}+1} \frac{1}{w^r}\, d_{\text{AGB}}^{(r)}\left(A_{\text{ext}}, B_{\text{ext}}\right), \tag{2.26}$$

where $\dfrac{1}{w^r}$ is a weighting factor that depends on the parameter $w > 0$.

The formation of blocks described above has the crucial disadvantage of dividing and further subdividing the images at the same position, thus creating a strong bias. Two pixels, one in the left half of A_{ext}, one in the right half of B_{ext}, will only be compared at the lowest resolution level (by means of contributing to the block mean), even if they are neighbouring pixels, like, for example, $a_{\frac{1}{2}2^{n_{\text{ext}}},\frac{1}{2}2^{n_{\text{ext}}}}^{(\text{ext})}$ and $b_{\frac{1}{2}2^{n_{\text{ext}}}+1,\frac{1}{2}2^{n_{\text{ext}}}}^{(\text{ext})}$. In contrast, $a_{\frac{1}{2}2^{n_{\text{ext}}},\frac{1}{2}2^{n_{\text{ext}}}}^{(\text{ext})}$ and $b_{\frac{1}{2}2^{n_{\text{ext}}}-1,\frac{1}{2}2^{n_{\text{ext}}}}^{(\text{ext})}$ will be compared at every resolution level, except for the

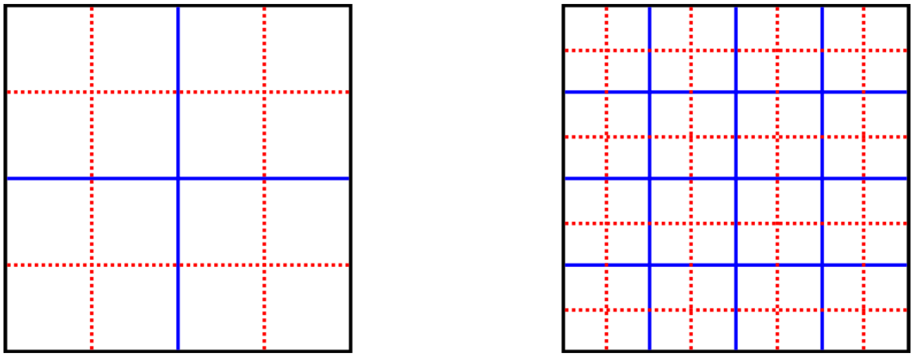


Figure 2.2: Original (blue, solid line) and alternate (red, dotted line) division of a $2^{n_{\text{ext}}} \times 2^{n_{\text{ext}}}$ image into blocks for resolution levels $r = 2$ (left image) and $r = 3$ (right image).

last. Thus, differences in A_{ext} and B_{ext} may affect $d_{\text{AGB}}\,(A_{\text{ext}}, B_{\text{ext}})$ differently, depending on their location.

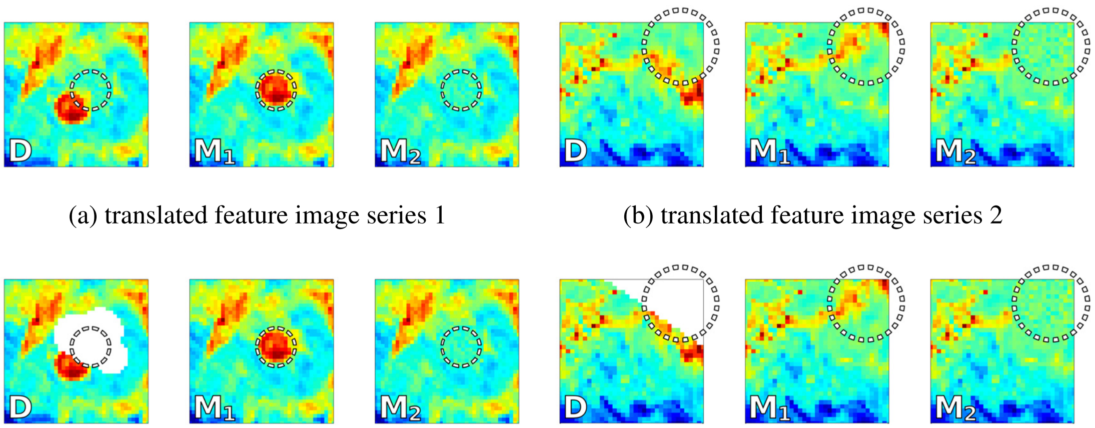
In order to decrease this bias, I introduce a second division into blocks for the resolution levels $r = 2, 3, \ldots, n_{\text{ext}}$. In this alternate division, the location of blocks is moved in X and Y direction by 2^{r-2} (half the blocks edge size; see Figure 2.2). The mean of the block distances at the original division and the alternate division is then used to compute $d_{\text{AGB}}^{(r)}$ for $r = 2, 3, \ldots, n_{\text{ext}}$ in my implementation.

By expanding A and B the complexity of the adapted grey block distance increased to $O(n_{\text{ext}}(2^{n_{\text{ext}}})^2)$ which can be expressed in terms of m and n as $O(\log_2(\max(m, n)) \times \max(m, n)^2)$.

2.4 Image Comparison Tests & Results

2.4.1 Test 1: Translated and Masked Features

In the first test, I explore whether the image comparison measures can detect translated and masked features in satellite images. For this purpose I introduce 3 images: D, M_1 and M_2. I assume that D is a satellite image that is compared to the model-derived images M_1 and M_2. In the first case, there is a feature that is present in both D and M_1, but at different locations; for example an eddy that is offset between data and model. This translated feature is not present in M_2 (see Figure 2.3 (a) and (b)). I consider it desirable for a distance measure d to rate D and M_1 to be closer than D and M_2, *i.e.*, $d(D, M_1) < d(D, M_2)$, because of the common feature in D and M_1 that does not appear in M_2. In this case I prefer the model to show, as opposed to *not* show, a feature (e.g., the eddy) that appears in



(a) translated feature image series 1

(b) translated feature image series 2

(c) masked feature image series 1

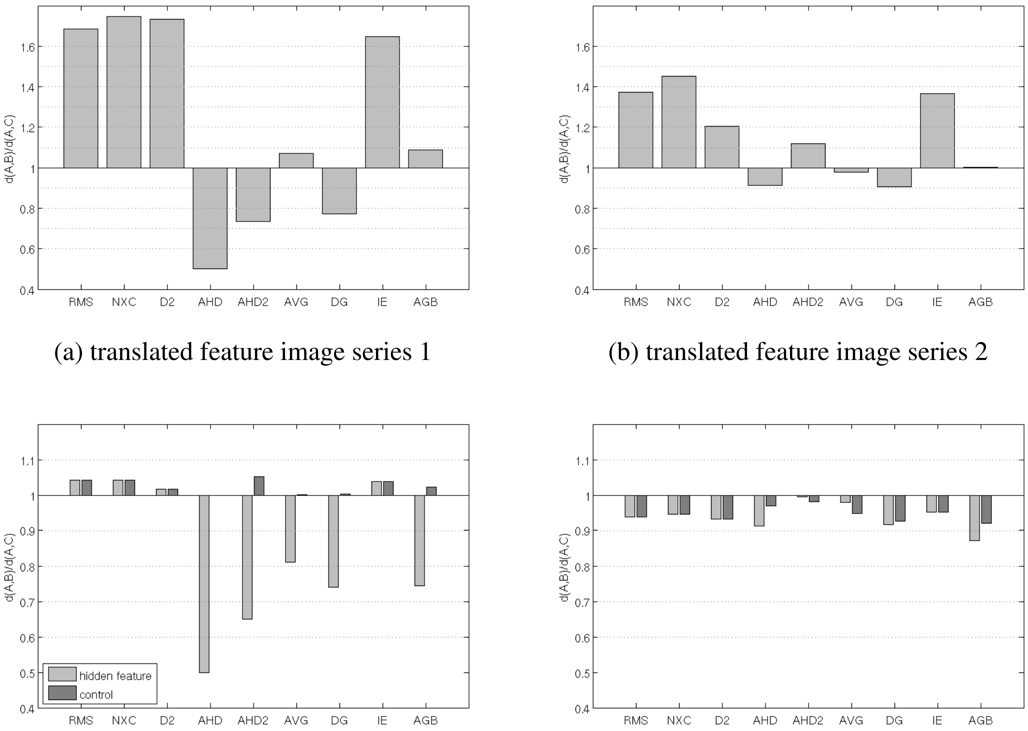
(d) masked feature image series 2

Figure 2.3: Examples of the manually created images used for the translated and the masked feature tests. In each test case, the first image represents the satellite image D which is compared to the images M_1 (center image) and M_2 (right image). D and M_1 share a common feature that does not appear in M_2. The location of this feature in M_1 is highlighted by a white dotted circle in every image. In the masked feature tests (c) and (d), this location is masked by *NaN*-values in D. The feature of interest in (b) and (d) is flipped upward from image D to image M_1.

the satellite image, even if not at identical locations.

I use slightly modified versions of the above sets of images for the masked feature test, in which the satellite image D contains missing values at the same location where M_1 contains the feature. Figure 2.3 (c) and (d) show examples of D, M_1 and M_2 for the masked feature case. The feature is masked in M_1 if only pixels in X_{real} are considered in the image comparison and therefore not accessible by pixel-by-pixel comparison measures. Yet, as both D and M_1 show the feature, I consider $d(D, M_1) < d(D, M_2)$ or the equivalent $\frac{d(D,M_1)}{d(D,M_2)} < 1$ a desirable characteristic of d.

I applied the image comparison measures to the test cases presented in Figure 2.3. The results are shown in Figure 2.4 as ratios of $\frac{d(D,M_1)}{d(D,M_2)}$. For the masked feature test, Figure 2.4 also includes the results of a control experiment, where the pixels with missing values in D are set to missing values in M_1 and M_2, so that $D_{NaN} = M_{1,NaN} = M_{2,NaN}$. I further added a small amount of noise to all images in Figure 2.3, so that none of the distances are zero and $\frac{d(D,M_1)}{d(D,M_2)}$ retains a meaningful value.



(a) translated feature image series 1

(b) translated feature image series 2

(c) masked feature image series 1

(d) masked feature image series 2

Figure 2.4: Results of tests for the image series shown in Figure 2.3, expressed as ratios $\frac{d(D,M_1)}{d(D,M_2)}$. Ratios smaller than 1 are desirable

In the cases with translated features, all pixel-by-pixel measures judge M_2 to be significantly closer to D than M_1, which is an inherent shortcoming of pixel-by-pixel comparison. For the neighbourhood-based measures the results are more diverse: AHD and Delta-g show the most desirable results and clearly rate D and M_1 closer than D and M_2, while the results of IE are similar to those of RMS. The other measures fall in between, with $\frac{d(D,M_1)}{d(D,M_2)}$ close to 1.

For the cases with masked features, the pixel-by-pixel measures show the same pattern: $\frac{d(D,M_1)}{d(D,M_2)}$ has the same value for the sets of images used in the test cases and their respective control experiments ($\frac{d(D,M_1)}{d(D,M_2)} \neq 1$ because not the entire feature is masked), the masked feature does not have any effect. The neighbourhood-based measures account for the masked feature to some degree, except for IE. For the masked feature image series 1 the distance of the two images is reduced significantly by the masked feature. For image series 2, this difference is not so obvious and the effect of the masked feature is insignificant or masked by noise.

The performance of the pixel-by-pixel measures is a direct result of not considering neighbouring pixels. This also leads pixel-by-pixel measures to ignore the pixels in D_{NaN} when comparing D to M_1 and M_2 in the masked feature case; consequently $\frac{d(D,M_1)}{d(D,M_2)}$ is the same for both, test and control experiment. All other methods compare pixels at different locations, which leads to an advantage in this test. Not all of the measures identify D to be closer to M_1, however. The individual results are also strongly dependent on the measures' scaling parameters, an influence which I will return to in Section 2.5. The results of the masked feature test show clearly that the adapted comparison measures can make use of pixels in $D_{\mathit{NaN}} \cap M_{1,\mathit{NaN}}$ and that these pixels can change results significantly. The exception is my implementation of IE which is not a pixel-by-pixel measure but also ignores all pixels in D_{NaN}.

In this test I focused on the very specific scenario of a single translated/masked feature. The results are somewhat artificial as the images have been specifically created to be nearly identical except for the feature of interest. Nevertheless, translated and masked features can easily be caused by inconsistencies in the model (e.g., time lags) in conjunction with large areas of missing values.

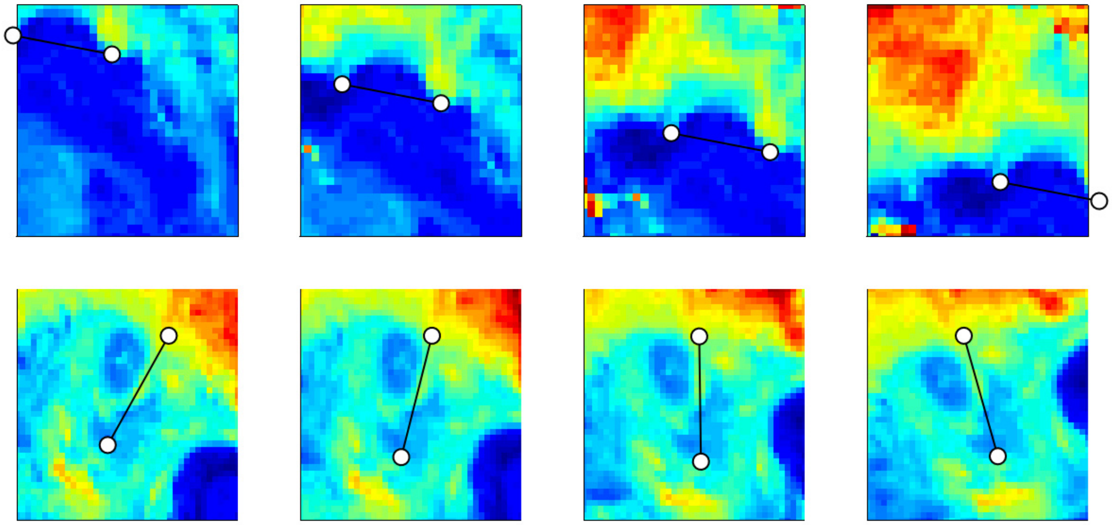


Figure 2.5: A series of 4 translated images (top row) and a series of rotated images (bottom row) used in Test 2. Markers are inserted into the images to illustrate the direction of the translation and the angles of the rotation, respectively.

2.4.2 Test 2: Translation & Rotation of Images

In this test I examine the effect of translation and rotation of a base image on the comparison measures. Stepwise translation or rotation of an image offers a simple way to create a series of similar images. Due to the nature of satellite images (as opposed to, e.g., images of white noise) an increased translation or rotation leads to an apparent increase in distance to the untransformed image. The image comparison measures should reflect this increase in distance.

For this test I generated a large number of series of transformed images. Starting with a large satellite image, each series was created by successively translating or rotating the image and then clipping it at the same position after each transformation. The clipping was done to ensure that the series contains no missing values that are introduced to the large image by the respective transformation. By adopting this approach I generated 100 series of translated images and 100 series of rotated images, each series containing between 5 and 6 images. Image sizes are constant within a series but range between 30×30 and 50×50 pixels among different series. The translations are performed along the X and Y axes as well as along their bisecting line. The translation distance between two images is 5 to 10 pixels. For each series of rotated images, the center of rotation is roughly in the center of the clipped image and the rotation angle between two images is 20°. In this test

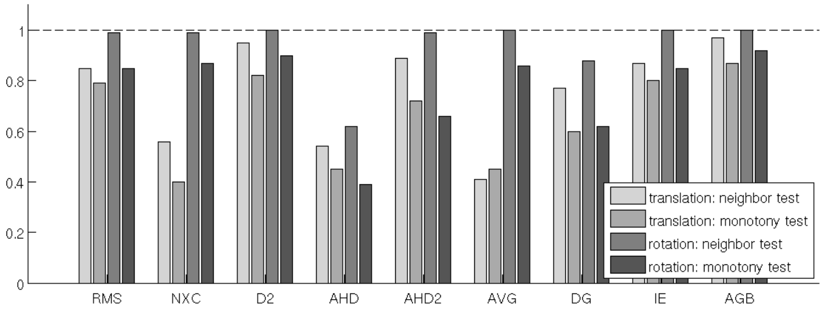


Figure 2.6: Fractions of passed *neighbour tests* and *monotonic tests* for 100 series of translated and rotated images.

the images do not contain missing values and I do not add any noise to the images.

Given a series of increasingly translated or rotated images $A_1, A_2, \ldots, A_q$ (e.g., see Figure 2.5) I perform two tests for each image comparison measure d:

neighbour test: A comparison measure d passes the *neighbour test* if any image in a series is closest to one of its neighbours. More precisely, the test is passed if for any given image in the series, the distance to one of the neighbouring images in the series is smaller than the minimum distance to any non-neighbouring image, *i.e.*, if

$$\min_{j\in\{i-1,i+1\}} d(A_i, A_j) < \min_{j\in\{1,\ldots,i-2\}\cup\{i+2,\ldots,q\}} d(A_i, A_j) \text{ for } i = 1, 2, 3, \ldots, q \quad (2.27)$$

monotonic test: d passes the *monotonic test* if, for the first image in the series, the distance to the other images in the series is monotonically increasing, *i.e.*, if

$$d(A_1, A_2) < d(A_1, A_3) < d(A_1, A_4) < \ldots < d(A_1, A_q). \quad (2.28)$$

The results of the two tests applied to both sets of images are shown in Figure 2.6. The ratio of passed tests is high for most distance measures, especially in the rotation *neighbour test* scenario. Generally, the results are better for the rotated images. This is especially obvious for AVG and NXC which have relatively low scores for the translated images, but perform roughly twice as well in the rotation test cases.

There is no indication that the pixel-by-pixel measures perform worse than the neighbourhood-based methods, in fact RMS and D2 are among the best performing measures in all 4 test cases. It is interesting to note that the results of D2 are slightly better than those of the RMS. Among the neighbourhood-based measures, AGB and IE perform well throughout all tests. The difference in parametrization among the two Hausdorff based measures is obvious, with AHD2 performing significantly better than AHD.

2.4.3 Test 3: Noise Sensitivity

In real life applications, it is to be expected that the satellite images contain some noise which may affect the image comparison. It is therefore important to test the noise sensitivity of image comparison methods. A multitude of publications have addressed this issue for regular grey scale images (e.g., *Avcıbaş et al.*, 2002). To assess how variable levels of noise in images affect the satellite image comparison methods, I examine how the results from the previous test change under the influence of noise.

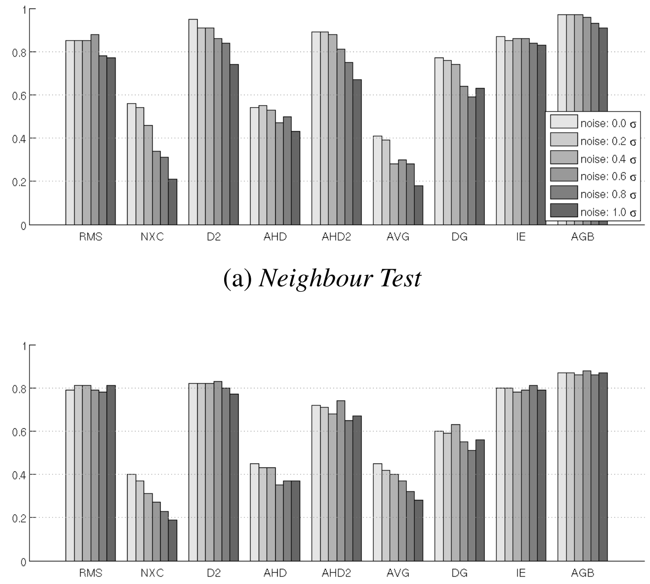
I used the 100 series of translated satellite images from the previous test. For each series, I determined the standard deviation σ of intensity values among all the images in the series. I then added Gaussian noise with mean 0 and standard deviation $x\sigma$ to each image (where adding the noise created negative values these were set to 0), where x is increased from 0.2 to 1.0 in increments of 0.2. For each noise level I performed the *neighbour* and *monotonic test* in the same manner as in the previous Test 2.

The results show that all comparison measures are affected by increased noise (Figure 2.7). The measures that employ averaging (AGB, IE and RMS) are the least sensitive to noise, with AGB performing best in both test cases. Especially the performance of AVG and NXC drops significantly as noise increases. D2 performed better than RMS at no noise, but drops below RMS with increasing noise. AHD and AHD2 exhibit a similar decline in performance compared to the no-noise results.

2.4.4 Test 4: *NaN*-Sensitivity

In addition to noise, missing values may also affect the performance of image comparison measures. In this test I examine the sensitivity of the image comparison measures to the location and number of missing values.

For this test I used two images of dimension 40×40 and with a translation distance of 10 pixels from a series of translated images in Test 2. I then created 100 copies of the second image and selected a fraction x_{NaN} of each copy to be missing values, varying x_{NaN} from 0.1 to 0.9. The selection was performed in a way that creates uniformly distributed



(a) *Neighbour Test*

(b) *Monotonic Test*

Figure 2.7: Fraction of passed *neighbour tests* and *monotonic tests* for different levels of noise.

circles of missing values in the image, forming cloud-shaped areas (to mimic regions of missing values on optical ocean imagery), see Figure 2.8 for examples. Finally, I use the image comparison measures to compare the first image to all of the *NaN*-covered copies of the second image, computing 100 image distances for each comparison measure. I then record mean and standard deviation of distances (Figure 2.9).

The mean can be thought of as a measure of bias in this test: For a single realization of a randomly *NaN*-covered image, I can expect that a major feature is covered resulting in an increase or decrease of distance when compared to another image. While this makes it more likely that the distance diverts further from the mean as the number of missing values is increased, I do not expect the mean to change significantly with an increase in x_{NaN} if the distance measure is unbiased. A significant change in mean implies that the distance measure is directly affected by the number of missing values and thus biased.

Noteworthy in this case are the adaptations IE and AVG which both show a significant bias. For IE, there is negative bias as the amount of missing values increases, lowering the mean distance. AVG, on the other hand, shows a positive bias, while the mean in all other distance measures is affected insignificantly.

The standard deviation in this test is a measure of missing value stability (*NaN*-stability).

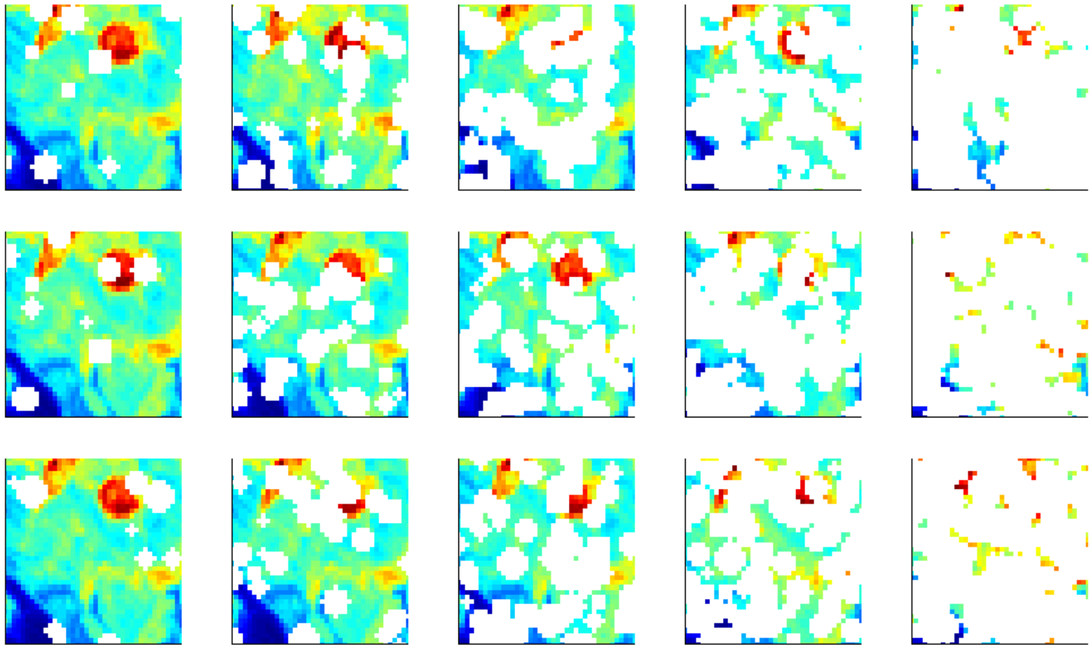
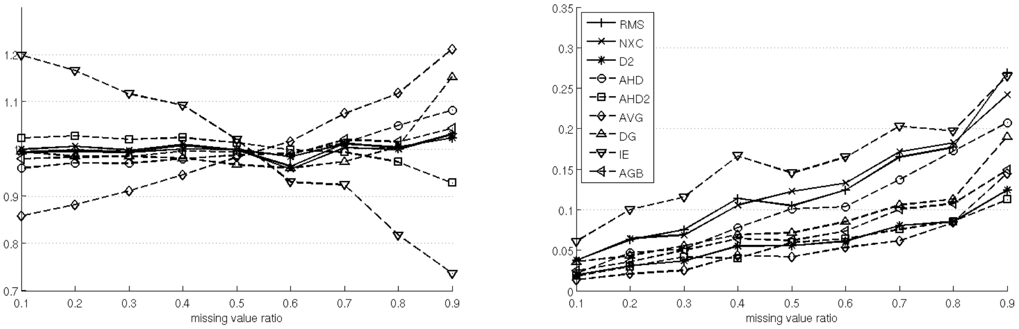


Figure 2.8: 3 realizations (rows) of the same image covered randomly in missing values. From left to right the number of missing values is increased in each column from 10% to 90% in increments of 20%.



(a) normalized mean distance

(b) standard deviation of distance

Figure 2.9: Mean and standard deviation of image distances for different levels of missing values. For each distance measure, the mean values have been normalized, so that their mean is 1.

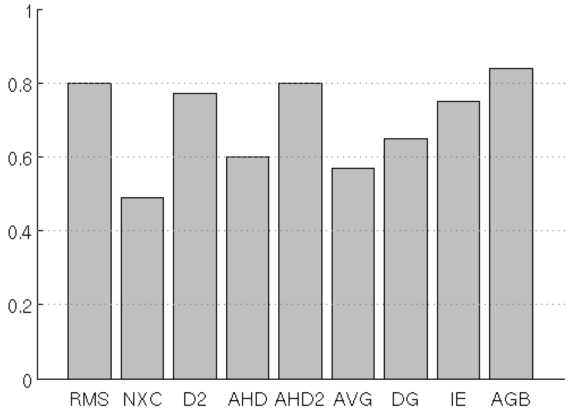


Figure 2.10: Fraction of passed *monotonic tests* for the *NaN*-translation case.

A lower standard deviation is desirable. It means that the distance between two images is less dependent on the location of missing values. The least stable distance measures, with highest standard deviations are IE, RMS and NXC. Measures that ignore pixels in $A_{NaN} \cap B_{real}$, with the exception of D2 which features one of the highest *NaN*-stabilities, show the worst performance in this test. They are followed by AHD which is significantly less *NaN*-stable than AHD2, illustrating the positive effect of its parametrization.

2.4.5 Test 5: *NaN*-Translation

In the previous test I assessed the sensitivity and stability of the image comparison measures when faced with missing values. Here, I examine the effects of missing values on the measures' ability to correctly estimate relative distances between images.

I use again 100 series of translated images and randomly add missing values to the images, as described for Test 4. Using the *monotonic test* I compare the untouched image with the altered ones to test if the distance in the series increases. Because the previous results suggest that this test is significantly harder to pass than the *monotonic test without missing values* (Test 2) or under the influence of noise (Test 3), the image series in this test are slightly less translated (3 to 5 pixels in between individual images).

The results of this test (Figure 2.10) are similar to those of previous tests: my implementations of AGB, AHD2, RMS, D2 and IE perform well. NXC is the worst, passing less than half of the tests.

2.4.6 Test 6: Time-Series

All of the previous tests focused on the comparison of images that were generated by translating or rotating a base image or one of its features. In this final test I consider a numerical ocean model-generated time series of images. A time series poses a somewhat different challenge to the series of images generated through translation: Features appear and disappear in the center of the image (e.g., by upwelling) and features such as fronts and eddies not only change their location, which is the case with translated images, but also their shape and size. This test is closer to a realistic application than the previous ones.

I use images of model-simulated chlorophyll (taken from the model presented in *Fennel et al.*, 2008). Using a model to generate test images is advantageous because there are no missing values, no noise and the time interval between images can be selected. Also, I expect the images to exhibit similar features to satellite images.

For this test, I use 6 series of 31 images (with a time difference of 12 days between two consecutive images); the images are clipped and ranged in size from 40×50 pixels to 70×150 pixels among different series (Figure 2.11). Since the apparent distance of two images in a time series does not necessarily grow monotonically with time, I use a variation of the *neighbour test* in this scenario: I compare the distances of an image to its two neighbours with the distances to the other images within a window of 7 images, instead of all images (as done in Test 2). Performing a *neighbour test* for every image series, I obtain a ratio of passed to total tests for each series. The mean ratio and its standard deviation among the series are displayed in Figure 2.12 for different noise and missing value levels, which were added as described in Test 3 and 4, respectively.

The best performers in the test cases without noise or missing values are AVG, D2 and AGB. The results of AGB are only slightly affected by the addition of noise and missing values, while the performance of D2 and especially AVG drops considerably with increasing noise. Among the pixel-by-pixel measures, D2 achieves better results than RMS in all cases. The relatively good results of NXC indicate that there is a high level of correlation for neighbouring images in the time series. For neighbourhood-based measures, many results confirm previous observations: The parametrization for AHD2 performs better than the one used in AHD. IE shows good performance that is not very sensitive to noise.

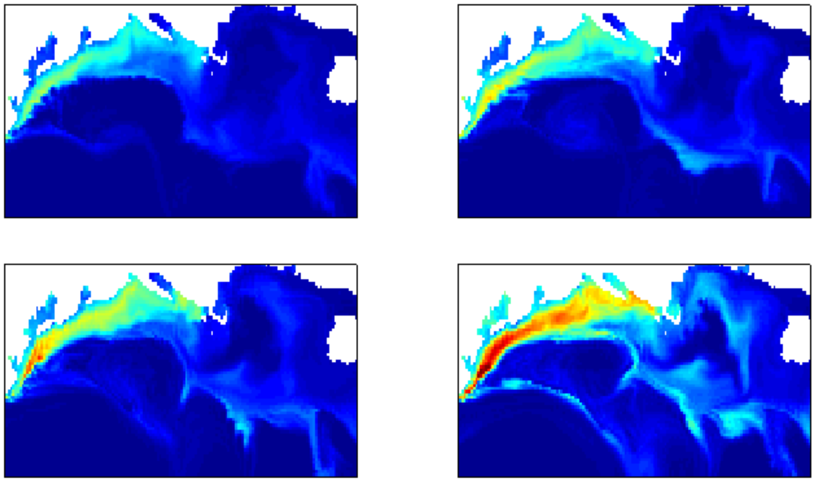


Figure 2.11: An extraction of 4 consecutive images in a time series of images generated by a physical-biological ocean model.

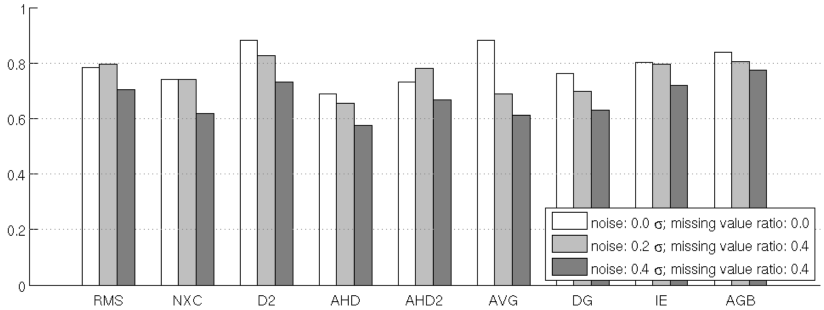


Figure 2.12: Results of the time-series test. Bar height indicates average fraction of passed tests for the 6 time series of images.

2.5 Discussion

For this chapter, I adapted 8 image comparison measures for use with satellite images. My motivation was to compare satellite imagery with spatial fields derived from numerical ocean models, which has applications in model skill assessment and data assimilation. The distinguishing features of satellite images compared to regular images are that their values are continuous non-integer values and that pixels or portions can be missing. The examined comparison measures range from simple but widely used pixel-by-pixel methods to more complex methods that evaluate the distance of pixels at different locations. I compared the behaviour of the comparison measures in 6 tests and assessed their response to different levels of noise and missing values. The qualitative performances of the measures are

Test	RMS	NXC	D2	AHD	AHD2	AVG	DG	IE	AGB
Test 1: translated & masked features	–	–	–	+	+	○	+	–	+
Test 2: translation & rotation	+	–	+	–	+	○	○	+	+
Test 3: noise sensitivity	+	–	+	–	○	–	–	+	+
Test 4: *NaN*-sensitivity	–	–	+	–	+	+[a]	○	–[a]	+
Test 5: *NaN*-translation	+	–	+	–	+	–	○	○	+
Test 6: time-series	○	–	+	–	○	○	–	○	+

[a] The significant bias introduced through missing values is not included in the rating.

Table 2.2: Qualitative performance ratings of the image comparison measures for the 6 Tests in Section 2.4. The symbols indicate relative performance in the tests: + is good performance, ○ is average performance and – is bad performance.

summarized in Table 2.2, and discussed in more detail below. One of the main goals was to compare the performance of the two very commonly used distance measures root mean square error and image correlation to lesser known alternatives.

Two measures outperformed both root mean square error and correlation throughout all of my tests. AGB, a neighbourhood-based measure, achieved the best overall test results. It is especially insensitive to noise and missing values. The pixel-by-pixel measure D2 also achieved better results than the two commonly used methods, but is less advantageous in cases where translated or masked features play an important role (compare Test 1). In applications where runtime is not a primary concern, I recommend using AGB as distance measure. If low runtime is required, D2 and RMS are good options. The ultimate choice should be application dependent.

Test 1 highlights the advantages of incorporating neighbouring pixels into satellite image comparison; while the neighbourhood-based methods can make use of the information in pixels that have values in the model but missing values in the data, the pixel-by-pixel measures cannot. Despite these inherent problems of pixel-by-pixel comparisons, the neighbourhood-based measures are not always superior to the pixel-by-pixel measures. In Test 4, D2 distinguished itself from the other pixel-by-pixel measures by exhibiting a very high *NaN*-stability.

One important factor to consider is the parametrization of the image comparison measures. Every neighbourhood-based measure I tested features one parameter or more. All of them contain a scaling parameter that weights the spatial distance of two pixels compared to their distance in intensity (the scaling parameters are listed in Table 2.1). Scaling parameters have a strong effect on the distance measures and the optimal choice depends on the resolution of the images that are compared. For high-resolution images, the distance between two neighbouring pixels is low and spatial distance needs to be weighted lower in comparison to distance in intensity than for similar images with coarser resolution. A possible explanation for the good results of AVG in Test 6 in contrast to the average results in the other tests is that its scaling parameter matches the scale of model-generated images from Test 6 better than the scale of the satellite images used for tests 1–5.

Beside the scaling parameters, other parameters have significant effects on the results, too. A good example is the performance of AHD and AHD2, two parametrizations of the adapted Hausdorff distance. AHD2 performed better than AHD in all tests, due to a change of one parameter (k_{max}) that controls the adapted Hausdorff distance's proneness to extreme values.

A larger number of parameters allows for a high level of customization but has the negative side effect that parameters may need to be adapted for each application. Root mean square error and normalized cross-correlation have no parameters and thus do not require the user to select a specific parametrization. The customizability of the neighbourhood-based measures varies strongly. The image Euclidean distance has only one parameter. The Hausdorff distance is highly customizable: *Dubuisson and Jain* (1994) present 24 variations of the Hausdorff distance, some with their own parameters. Pixel-by-pixel measures can also have a high number of parameters, e.g. there are various flavors of entropic distances. The entropic distance D_2 used here is just one of 6 different entropic distances introduced in *Di Gesú and Roy* (2002), each can be customized further.

Missing values are one of the distinguishing features of satellite images. Test 4 focused directly on the effects of the number of missing values on image distances. Bias introduced through missing values has different effects: In model skill assessment or data assimilation, bias is not an important issue since the number of missing values stays constant; one satellite image is compared to a number of model-derived images, each with the same number of missing values. Bias becomes important in scenarios where one image is compared to two or more images that have different number of missing values. *NaN*-stability, on the other hand, plays a more universal role and my results indicate that even for low ratios of missing values, the comparison measures display significantly different levels of *NaN*-stability. Here lies one of the apparent weaknesses of the standard pixel-by-pixel measures RMS and NXC: very low *NaN*-stability.

All of my tests are based on the assumption that I can produce images that are less similar to one another by either moving a specific feature within the image, by increasing translation or rotation of the entire image, or by increasing the time difference of images in a time-series. This is done for a reason; apart from time-series and simple transformations, there is no easy way to create a set of images for which there is an objective indication of image similarity. Using any comparison measure as the standard for determining image similarity would be circular and predetermine the outcome of the tests.

Beside test performance, ease of implementation and runtime need to be considered. The pixel-by-pixel measures are generally very easy to implement and they are the fastest methods. Because they compare only pixels at the same location they have a complexity $O(mn)$. All neighbourhood-based measures have a higher complexity and, due to their more elaborate means of comparing pixels at different locations, are also harder to implement and more costly to compute. The variation in runtime and ease of implementation among them is significant. One of the fastest neighbourhood-based measures I tested is AGB. It has a complexity of roughly $O(\log_2(\max(m,n))\max(m,n)^2)$. AVG is the only other neighbourhood-based measure with a complexity below $O((mn)^2)$, although its runtime is strongly dependent on its parametrization. The slowest methods are AHD, IE and DG; runtime of the latter is also very dependent on the choice of parameters. Ease of implementation is not as easy to judge objectively. I found image Euclidean distance and adapted Hausdorff distance to be the easiest to implement while an efficient implementation of average distance, adapted grey block distance and especially Delta-g took more effort.

2.6 Conclusions

This chapter shows that low level image comparison measures, developed for regular grey scale images, can successfully be adapted to work with satellite images. In comparison to standard comparison measures such as root mean square error (RMS) and cross-correlation, two of the adapted measures show better performance throughout all of my tests. These measures are the adapted grey block distance (AGB), which compares images at multiple scales, and the entropy based measure D_2. The advantages of these measures are especially apparent in scenarios that involve missing values, one of the distinguishing features of satellite images. AGB also exhibits the lowest sensitivity to noise among the measures I tested and I would therefore recommend it over RMS. While AGB requires more runtime than pixel-by-pixel measures, it is among the fastest neighbourhood-based methods I tested. In cases where runtime or ease of implementation are the prime concerns, the pixel-by-pixel measure D_2 represents a viable alternative to RMS.

Chapter 3

Estimating Time-dependent Parameters for a Biological Ocean Model Using an Emulator Approach[1]

3.1 Introduction

Simple models are often considered advantageous over more complex ones, because they tend to be easier to interpret and to calibrate and less expensive computationally. Their low complexity is typically achieved by combining many properties of the simulated system into single model variables and averaging them in time and space. In the context of biological ocean models, a good example for this is the blending of many plankton species into functional groups or often even into bulk model variables for phytoplankton and zooplankton (so called NPZD-class models). In the bulk variable treatment, each variable represents a large variety of real species with a range of specific physiological characteristics (e.g. different growth and nutrient uptake rates, different carbon-to-chlorophyll ratios). Since the abundance of these species and their relative contribution to the plankton community changes in space and time, so should the physiological characteristics of the bulk variables. In this chapter, I find evidence for temporal and spatial dependence of the parameters of a biological model that contains just two plankton variables, suggesting that using static parameters is overly simplified and suboptimal. Using an emulator approach in combination with a temporally and spatially dense set of satellite observations I can

[1] based on: Mattern, J. P., K. Fennel, and M. Dowd, Estimating time-dependent parameters for a biological ocean model using an emulator approach, *Journal of Marine Systems, 9697*, 3247, 2012.

effectively infer parameter values that evolve in time and space and lead to an improved representation of plankton in the model.

Many studies have employed data assimilation in the context of biological models, often in order to optimize the poorly known parameters but also to update the model state and improve the models' forecast abilities. The techniques used in these studies can be divided into 3 broad categories: (1) Variational techniques, such as 3DVAR and 4DVAR (e.g. *Smedstad and O'Brien*, 1991; *Lawson et al.*, 1996; *Powell et al.*, 2008), (2) Monte-Carlo based techniques which include the ensemble Kalman filter (e.g. *Allen et al.*, 2003; *Evensen*, 2003; *Hu et al.*, 2012), particle filter methods (*Losa et al.*, 2003; *Mattern et al.*, 2010a; *Dowd*, 2011) and Markov chain Monte Carlo methods (e.g. *Dowd*, 2007; *Jones et al.*, 2010), and (3) emulator techniques. Emulators differ from the aforementioned techniques in that they effectively replace computationally expensive model simulations with fast approximations. Emulators require a set of model simulations for specific values of the stochastic inputs (parameters), then approximate unknown model output based on these simulations. The approximation is used in place of the true model output, eliminating the need for additional model simulations. This property makes emulators more efficient than other approaches, especially Monte Carlo techniques which rely on ensemble generation through random sampling and therefore generally require considerably more model simulations (*Rougier and Sexton*, 2007).

The emulator approach that I use in this chapter is based on the polynomial chaos expansion, which was first introduced by *Wiener* (1938) and later extended (*Askey and Wilson*, 1985; *Wan and Karniadakis*, 2006). Polynomial chaos relies on a set of orthogonal polynomial basis functions for the approximation of model results. The method has been applied widely in physical sciences (see *Xiu and Karniadakis*, 2003, for an overview), with only few applications in an oceanographic context (*Lucas and Prinn*, 2005; *Thacker et al.*, 2012). Emulators applied in oceanographic contexts include emulators based on Gaussian process models (*Scott et al.*, 2011) and other techniques (*Frolov et al.*, 2009; *Hooten et al.*, 2011). To my knowledge, emulator approaches have been used in the context of biological ocean models in only one study by *Hooten et al.* (2011) where 7 biological parameters are estimated. I focus this chapter on just 2 biological parameters, but employ the emulator to estimate their time-dependence in order to achieve a better representation of plankton dynamics in the model and an enhanced understanding of the biological model dynamics.

I further use the emulator to obtain improved state estimates in an efficient manner.

Previously, two approaches have emerged to better represent the diversity of plankton. The first approach is to divide planktonic species into functional groups so that each plankton variable represents a more homogeneous and functionally distinct group of fewer species. While the simpler NPZD-class models contain only one phytoplankton and one zooplankton variable (e.g. *Doney et al.*, 1996; *Franks and Chen*, 1996; *Fennel et al.*, 2008), many biological models include two or more phytoplankton variables distinguishing, for example, between small and large phytoplankton, diatoms, diazotrophs etc. (e.g. *Moore et al.*, 2001; *Aumont et al.*, 2003; *Gregg et al.*, 2003; *Lehmann et al.*, 2009). One obvious limitation to adding more and more functional groups is that the number of poorly known parameters necessary for describing the biological interactions between functional plankton groups, and different pools of other organic and inorganic matter, increases dramatically (*Denman*, 2003) with consequent degradation of predictive skill.

In a recent, alternative approach, *Follows et al.* (2007) initialized a model with roughly 100 phytoplankton groups with their functional parameters drawn randomly from prescribed probability distributions. This approach allows for spatial and temporal variations in the self-organizing plankton community structure that emerges from local environmental conditions and competition (*Goebel et al.* (2010), see also review by *Follows and Dutkiewicz* (2011)) and represents a significant step toward a more flexible and realistic representation of plankton diversity in biological models. One drawback may be the large computational overhead required to carry on the order of 100 state variables.

I propose an alternative approach for the simulation of functional groups in biological models, namely incorporating variability or uncertainty by allowing the plankton parameters to be random variables with probability distributions that are allowed to vary in time. The main idea is that a small number of variables can achieve a more flexible representation of the plankton community, if their parameters are not fixed but stochastic properties governed by probability distributions. This approach effectively allows one phytoplankton variable to take on a range of different growth rates, sinking rates, etc. mimicking the behaviour of different functional groups at different times. In combination with observations and a data-assimilative framework, the uncertainty in the model can be constrained by limiting the stochastic parameters to ranges that explain the observations best. I accomplish this using the emulator approach described above.

Most studies which combine biological modelling with the estimation of stochastic parameters treat influences such as the varying plankton assemblage as error terms (*Dowd*, 2011). In these cases one aims to find a static distribution for the stochastic parameter of interest. Stochastic parameters then induce uncertainty into the model state; the mean (or median) model state represents the best estimate of the true state, while its variance (or error estimate) captures the model uncertainty including the variations caused by changing plankton assemblages. Here, my approach is different: using a time-series of observations, I find the parameter values that best explain each observation. That is, parameter values are allowed to change in time and the best state estimate is the model state associated with the series of time-varying parameters.

For this purpose, I use a set of daily chlorophyll satellite images to obtain daily values for two parameters of the biological model. I find that there is a strong time-dependence in the optimal parameter values which follow a seasonal cycle. Chlorophyll state estimates derived from the time-varying parameters are significantly closer to observed chlorophyll values than those of a model simulation with optimized fixed parameters. The improvement remains significant in a cross-validation experiment which I performed to avoid overfitting the observations. This is evidence that the introduction of time-varying parameters can achieve a more realistic representation of the biological dynamics in a typical biological ocean model.

3.2 Methods

3.2.1 Biological Model and Parameters of Interest

The model domain is the Middle Atlantic Bight (MAB), a coastal region in the northwest Atlantic that stretches from Cape Cod in the north to Cape Hatteras in the south (Figure 3.1). The model is based on the Regional Ocean Modeling System (ROMS; *Haidvogel et al.* (2008)) and consists of a physical model coupled with a biological component. Open boundary conditions for temperature, salinity, sub-tidal frequency velocity and sea level are taken from the larger-scale MAB and Gulf of Maine (MABGOM) regional model described in *Chen and He* (2010). Further details of the physical model are described in *Hu et al.* (2012). The biological component is described in *Fennel et al.* (2006); it simulates a simplified nitrogen cycle and has been employed successfully in various modelling studies (*Fennel et al.*, 2008; *Fennel and Wilkin*, 2009; *Previdi et al.*, 2009; *Bianucci et al.*, 2011).

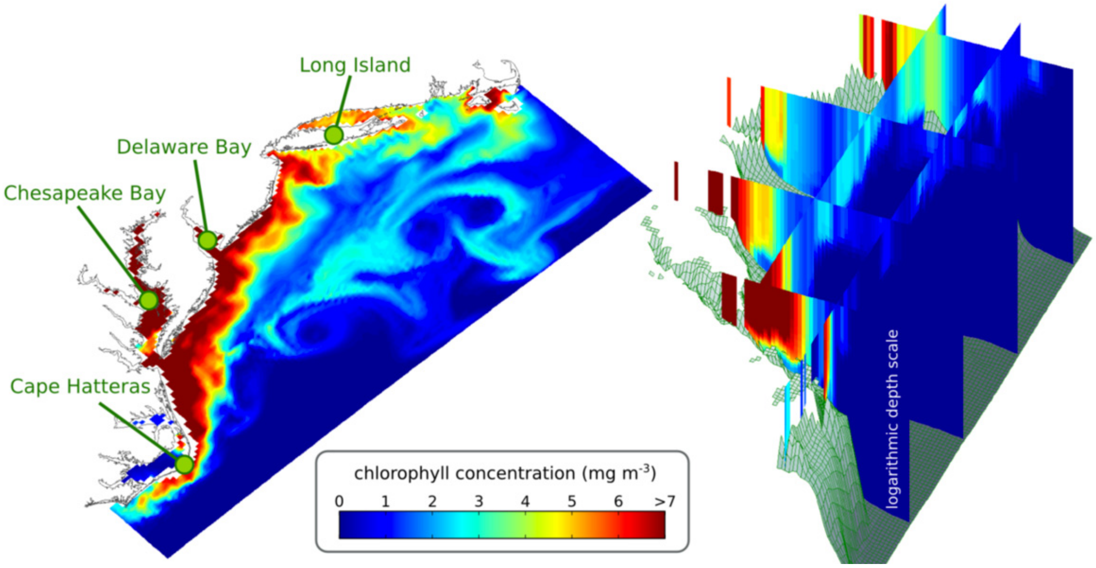


Figure 3.1: Snapshot of the chlorophyll variable in the biological model. The left image shows the surface concentration, on the right multiple slices of the 3-dimensional chlorophyll field are placed over the bathymetry of the model region.

The model contains one state variable each for phytoplankton and zooplankton, as well as variables for chlorophyll, nitrate, ammonium and small and large detrital nitrogen. Chlorophyll is simulated separately from phytoplankton to account for the effects of photoacclimation which allows phytoplankton species to regulate their chlorophyll content based on the availability of light and nutrients (*Geider et al.*, 1998). Here, all model runs are for 1 year, starting on 1 January 2006 and ending on 31 December 2006. The initial and boundary conditions for the biological variables are taken from a larger scale model of the Northeast North American (NENA) shelf that uses the same biological component (*Fennel et al.*, 2006) as described in *Hu et al.* (2012). A spin-up time of 12 days is used before the model is confronted with observations. The start of the simulations in winter, a time when variation of the biological parameter values has little impact on the model state (see results in Chapter 5) justifies the relatively short spin-up time.

Despite the relative simplicity of the biological model with only two plankton variables, one for phytoplankton and one for zooplankton, it requires more than 30 physiological parameters for the biological dynamics. Here I focus on only two of these parameters: the maximum ratio of chlorophyll to phytoplankton carbon, and the maximum grazing rate of zooplankton. These two parameters were selected based on a sensitivity study where I compared the effect of variations in several candidate parameters on the chlorophyll field. Specifically, I performed 1-year simulations for a baseline parameters set and for parameters sets where one parameter was doubled and halved. The selection criterion is based on chlorophyll sensitivity because I use chlorophyll satellite observations (see Section 3.2.2 below).

The most sensitive parameters were found to be the maximum ratio of chlorophyll to phytoplankton carbon and a parameter controlling the grazing rate of zooplankton. From here onward, I will refer to the maximum ratio of chlorophyll to phytoplankton carbon as θ_1 and the maximum grazing rate of zooplankton as θ_2. The physiological parameter θ_1 sets an upper limit on the concentration of chlorophyll relative to phytoplankton biomass. In the model equations (*Fennel et al.*, 2006), the fraction of phytoplankton growth that is devoted to chlorophyll synthesis, ρ_{Chl}, is a function of θ_1:

$$\rho_{\text{Chl}}(\theta_1) = \theta_1 \, \frac{\mu \, X_{\text{Phy}}}{\alpha \, I \, X_{\text{Chl}}}.$$

Here, X_{Phy} and X_{Chl} are the phytoplankton and chlorophyll variables respectively and $\frac{\mu \, X_{\text{Phy}}}{\alpha \, I \, X_{\text{Chl}}}$ is the ratio of achieved-to-maximum potential photosynthesis (*Geider et al.*, 1997). The parameter θ_2 controls the growth and abundance of zooplankton, which interacts with and is strongly dependent on the concentration of phytoplankton. It scales the zooplankton grazing rate g according to:

$$g(\theta_2) = \theta_2 \, \frac{X_{\text{Phy}}^2}{k_{\text{P}} + X_{\text{Phy}}^2},$$

where k_{P} is the half-saturation concentration of phytoplankton ingestion. The model equation that contains the sources and sinks of chlorophyll incorporates both $\rho_{\text{Chl}}(\theta_1)$ and $g(\theta_2)$, in its full form it is:

$$\frac{\partial X_{\text{Chl}}}{\partial t} = \underbrace{\rho_{\text{Chl}}(\theta_1) \, \mu \, X_{\text{Chl}}}_{\text{growth}} - \underbrace{g(\theta_2) \, X_{\text{Zoo}} \, \frac{X_{\text{Chl}}}{X_{\text{Phy}}}}_{\text{grazing}} - \underbrace{m_{\text{P}} \, X_{\text{Chl}}}_{\text{mortality}} - \underbrace{\tau \, (X_{\text{SDet}} + X_{\text{Phy}}) \, X_{\text{Chl}}}_{\text{aggregation}} \, .$$

Here X_{Zoo} and X_{SDet} are the zooplankton and the small detritus variables respectively; the constants m_{P} and τ are mortality and aggregation parameters. Since both θ_1 and θ_2 directly scale major growth and loss terms, it is not surprising that variations in either parameter have a strong effect on the chlorophyll concentration.

3.2.2 Chlorophyll Observations and Model-Data Comparison

Observations are essential in model calibration, optimization and validation. In all cases model output is compared to observations or made to fit the observations according to a chosen criterion. Thus both the observations and the choice of criterion can affect the results. In Chapter 2, I formulated and analyzed several criteria tailored to model-data comparisons of satellite observations, and suggested a new measure, the "adapted grey block distance" (AGB) as preferable over more commonly used measures such as the root mean square error. For the calculation of AGB, two images are compared at different resolution levels by dividing them into subsequently smaller, square blocks and determining the average intensity value for each block. For each resolution level, from the coarsest resolution where one block encompasses the entire image, up to the finest where each block is made up of a single pixel, the root mean square error is determined, weighted and summed, resulting in the AGB distance value. When comparing an image derived from the model to an observation image, the comparison at multiple resolutions can be advantageous when noise is present in the observations and there are spatial offsets in the images (*Mattern et al.*, 2010b). The AGB is also adapted to deal with missing values in images. Because of these qualities, I make use of AGB for the remainder of this chapter. Note however, that the methods described in this chapter do not require the use of AGB, and that any suitable model-data distance measure can be substituted.

The observations used in this chapter are daily images of surface chlorophyll concentrations derived from the SeaWiFS satellite for the year 2006 (350 images are available). Each image represents a daily average of one or more satellite scenes that have been interpolated onto the model grid. Due to clouds and other effects that impair the view of the optical satellite sensors, large portions of the images may be missing (compare, e.g., the sample satellite images in Figure 3.2). In addition, noise is present in the satellite data set and especially evident in coastal regions (see, e.g. the average chlorophyll development of the data in the estuaries in Fig 3.7). The same observational data set used here was also used in *Hu et al.* (2012) and is described in more detail there.

The distance value of AGB when comparing the satellite image at time index t with the corresponding model chlorophyll field is denoted as

$$\mathrm{d}(t, \theta_1, \theta_2) \text{ for } t = 1, 2, \ldots, n_{\text{obs}}. \tag{3.1}$$

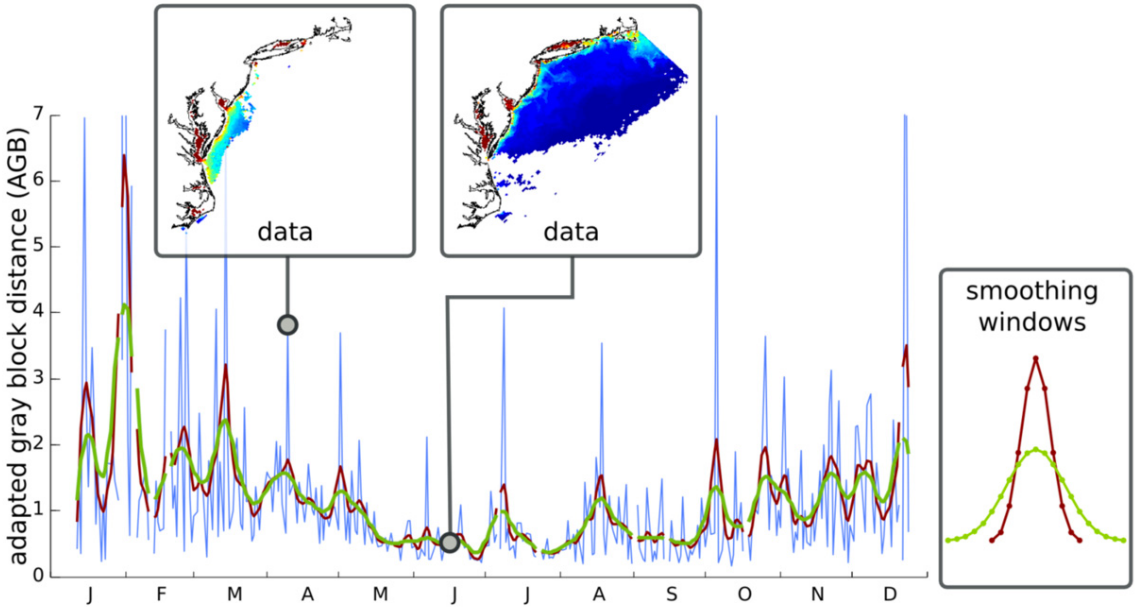


Figure 3.2: The time-dependent model-data distance function for the optimal fixed parameter set and two smoothed versions of it. The two smoothed curves correspond to smoothing intensities of 5 and 10 (dark red and green, respectively); the Gaussian smoothing windows are shown on the right. Two data images illustrate one point of the distance function with a high value and many missing values as well as one with a low distance function value and few missing values. High distance values tend to be caused by little available data.

Here, n_{obs} is number of (not necessarily equally spaced) time steps with available observations (in this case $n_{\text{obs}} = 350$). The dependence of d on the parameters θ_1 and θ_2 stems from the influence of both parameters on model chlorophyll.

3.2.3 The Emulator: The Polynomial Chaos Expansion

Polynomial chaos is an approach to quantifying how uncertainty in a model's inputs relates to uncertainty in its outputs. Like other emulator approaches it uses deterministic model runs given specific values of the uncertain inputs (i.e. the model's parameters, initial or boundary conditions, all of which will be referred to as parameters in the following). The resulting model output for these specific parameter values is then interpolated in parameter space to obtain approximations of the model output for all parameter values within the considered range. Since each uncertain input has a probability distribution (a prior distribution which must be specified) every model output that is dependent on the uncertain inputs must also have a distribution (induced by the uncertainty in the input). The polynomial chaos expansion provides a framework with which the properties of the distribution of any output value, such as the mean and variance of the distribution, can be approximated easily.

As the name suggests, polynomial chaos performs a polynomial interpolation in parameter space. This feature becomes useful in cases where one wants to obtain an estimate of the model output for a parameter value without performing additional model simulations. Using polynomial chaos, one can estimate any model output for the parameter values of choice. These outputs could range from the phytoplankton concentration in a given grid cell to the entire 3-dimensional chlorophyll field of the model. The interpolation feature of the polynomial chaos expansion can also be used to approximate other functions which depend on the uncertain inputs, e.g. I use it here to approximate the distance function in (3.1).

A short introduction to the polynomial chaos theory follows (for recent, more detailed studies see *Xiu and Karniadakis* (2003) and *Marzouk and Najm* (2009)). Since the focus is on stochastic parameters I base this explanation solely on parameter inputs. While I include two stochastic parameters, the methodology is described below for only one stochastic parameter θ. By assuming independent stochastic parameters, the theory translates in a straightforward manner into multidimensional parameter space (*Xiu and Karniadakis*, 2002).

Let the function $f(\mathbf{x}, t, \theta)$ be the property of interest, e.g. any model output or a function thereof (e.g. the distance measure in (3.1) which is a function of the model's chlorophyll output). The function f may be dependent on space $\mathbf{x}$, time t and the uncertain parameter θ. In the polynomial chaos expansion f is approximated by a basis function expansion:

$$f(\mathbf{x}, t, \theta) = \sum_{k=0}^{k_{\text{max}}} a_k(\mathbf{x}, t)\ \phi_k(\theta) \quad + \epsilon_{\text{trunc}}(\theta) \tag{3.2}$$

where $a_k(\mathbf{x}, t)$ are expansion coefficients, independent of the uncertain input θ, and the kth basis function $\phi_k(\theta)$ is a polynomial of order k in the parameter space defined by θ. The parameter k_{max} is the maximum order of polynomials used in the approximation and determines the quality of the approximation and ϵ_{trunc} is the truncation error. Without cutoff, i.e. for $k_{\text{max}} = \infty$, the approximation is exact and $\epsilon_{\text{trunc}}(\theta) = 0$. However, the number of required model runs grows with k_{max}, so that computational constraints enforce the use of relatively small values in typical applications.

k	$\phi_k(\theta)$ for $\theta \in [-1, 1]$	N_k
0	1	1
1	$1\,\theta$	$\frac{1}{3}$
2	$\frac{1}{2}(3\,\theta^2 - 1)$	$\frac{1}{5}$
3	$\frac{1}{2}(5\,\theta^3 - 3\,\theta)$	$\frac{1}{7}$
4	$\frac{1}{8}(35\,\theta^4 - 30\,\theta^2 + 3)$	$\frac{1}{9}$
5	$\frac{1}{8}(63\,\theta^5 - 70\,\theta^3 + 15\,\theta)$	$\frac{1}{11}$
6	$\frac{1}{16}(231\,\theta^6 - 315\,\theta^4 + 105\,\theta^2 - 5)$	$\frac{1}{13}$

Table 3.1: The first 7 Legendre polynomials ϕ_k and their associated normalization factors N_k.

The choice of polynomials in equation (3.2) is dependent on the probability density function of the parameter θ which is denoted $p(\theta)$. The polynomials are chosen to be orthogonal with respect to p, so that

$$\int_S \phi_k(\theta)\ \phi_i(\theta)\ p(\theta)\ \mathrm{d}\theta = \delta_{k,i}\ N_k\ . \tag{3.3}$$

Here S is the support of p (the region where $p(\theta) > 0$); the Kronecker delta function $\delta_{k,i}$ is equal to 1 if $k = i$ and 0 otherwise; $N_k = \int_S \phi_k(\theta)^2\ p(\theta)\ \mathrm{d}\theta$ is a normalization factor specific to the kth polynomial and independent of θ. All common distributions have well known sets of polynomial basis functions (*Xiu and Karniadakis*, 2002) and polynomial chaos can be generalized further to accommodate arbitrary distributions of θ (*Wan and Karniadakis*, 2006). For example, the corresponding set of orthogonal polynomials for a θ with uniform distribution, which I will use in this chapter (see Section 3.2.4), are the Legendre polynomials and ϕ_k is the kth Legendre polynomial. The first 7 Legendre polynomials and their associated normalization factors are listed in Table 3.1.

i	1	2	3	4	5	6	7
$\theta^{(i)}$	-0.9491	-0.7415	-0.4058	0	0.4058	0.7415	0.9491
ω_i	0.1295	0.2797	0.3818	0.4180	0.3818	0.2797	0.1295

Table 3.2: The quadrature points $\theta^{(i)}$ and associated weights ω_i for Gauss Legendre quadrature of maximum order $k_{\text{max}} = 6$.

To perform the basic polynomial chaos approximation in equation (3.2), one needs to compute the coefficients a_k. They are given by

$$a_k(\mathbf{x}, t) = \tfrac{1}{N_k} \int_S f(\mathbf{x}, t, \theta)\phi_k(\theta)\ p(\theta)\ \mathrm{d}\theta, \tag{3.4}$$

which is approximated by a Gaussian quadrature as (*Xiu and Karniadakis*, 2002):

$$a_k(\mathbf{x}, t) \approx \tfrac{1}{N_k} \sum_{i=0}^{k_{\text{max}}} f(\mathbf{x}, t, \theta^{(i)})\phi_k(\theta^{(i)})\ \omega_i. \tag{3.5}$$

Here $\theta^{(i)}$ is a quadrature point in parameter space and given by the roots of $\phi_{k_{\text{max}}+1}$ and the scalars ω_i are Gaussian quadrature weights (both are dependent on the choice of the distribution of θ and the parameter k_{max}). Table 3.2 contains the quadrature points and their weights for uniform θ and Gauss-Legendre quadrature with $k_{\text{max}} = 6$. From a computational perspective, it is important to note that the computation of the coefficients a_k requires the computation of $f(\mathbf{x}, t, \theta^{(i)})$ at each quadrature point $\theta^{(i)}$ for $i = 0, 1, \ldots, k_{\text{max}}$. In other words, $k_{\text{max}} + 1$ model runs are needed. Increasing the precision of the approximation by increasing k_{max} by one, therefore comes at the cost of an additional model run.

One advantage of polynomial chaos lies in the straightforward way in which the uncertainty in the input (the stochastic parameter θ) translates into the output (f). Due to the orthogonality of the polynomials, expected value and variance of f are straightforward to calculate once the coefficients a_k have been computed. Expectation and variance are given by

$$\begin{aligned}\mathbb{E}(f(\mathbf{x},t,\theta)) &= a_0(\mathbf{x},t),\\ \text{var}(f(\mathbf{x},t,\theta)) &= \sum_{k=1}^{n} a_k^2(\mathbf{x},t)\ N_k.\end{aligned} \tag{3.6}$$

They represent the mean and variance of f introduced by the variation of θ. To obtain good estimates of the full variance of f, e.g. for the purpose of creating estimates of model error, it is important to capture all the error of the uncertain inputs and to choose appropriate prior distributions for the inputs.

As mentioned, the above equations feature only one stochastic parameter θ. When expanded to more than one parameter, the computational cost for polynomial chaos increases exponentially with the number of stochastic parameters. For example, when including n_θ stochastic parameters to be approximated using polynomials of order k_{max}, $(k_{\text{max}}+1)^{n_\theta}$ model runs are required. Furthermore, if one desires to increase the order of polynomials, the quadrature points change, so that completely new model runs will have to be performed. However, it should be noted that the model simulations are only performed once prior to any attempts at inference.

3.2.4 Polynomial Chaos Setup and Approximation

When implementing polynomial chaos, the factors that need careful considerations are (1) the choice of uncertain model inputs (parameters), (2) the prior distributions assigned to these inputs, and (3) the highest order of polynomials k_{max} for each input. In an ideal scenario, one would take a fully Bayesian approach, that is treat all inputs that are not completely known as uncertain and incorporate them into the polynomial chaos procedure. However, complex models such as 3-dimensional ocean models have a large number of inputs that are not fully known, e.g. many parameters, physical forcing, boundary conditions, etc. To incorporate all these sources of uncertainty into the polynomial chaos expansion would necessitate a large number of model runs and prove to be infeasible using current computing resources.

Here, I undertake a targeted study focused on just two biological parameters. Once the uncertain inputs are selected, assigning a prior distribution to the inputs requires careful consideration, as one typically has little knowledge of the uncertainty (or error) of the

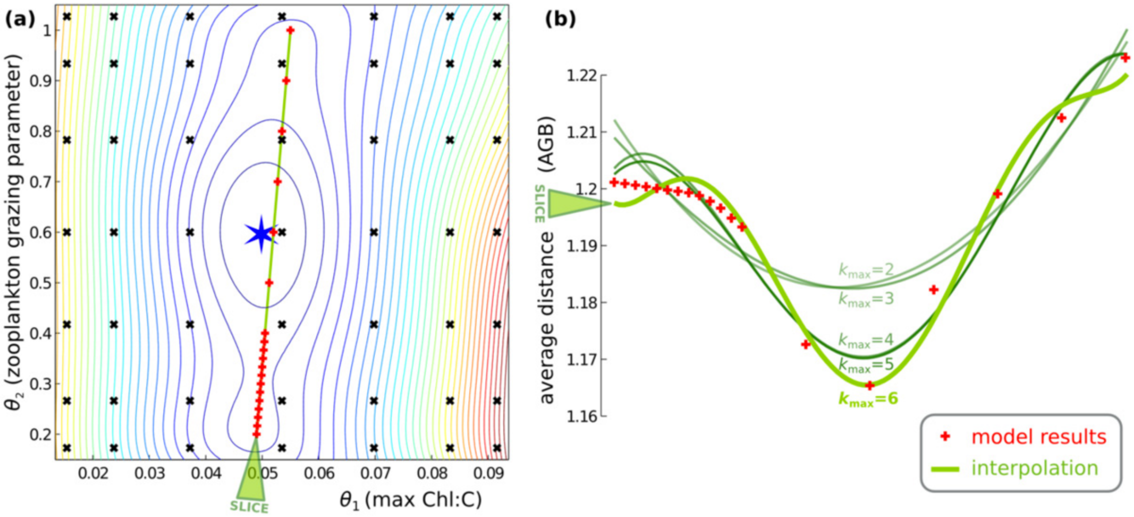


Figure 3.3: The interpolated time-averaged distance function. Image (a) shows a contour plot of the distance function with quadrature points (black crosses) and the global minimum (blue star) which corresponds to the optimal fixed parameter set. A slice through the interpolated function in (a) is shown in (b) as a green line in comparison to model results (red pluses). The slice in (b) also illustrates the effect of lowering k_{max}, thereby using fewer basis functions to approximate the average distance function.

inputs. Often, and the case here, one bases the prior distribution on previous experiments, literature values or educated guesses. In the choice of k_{max}, one is again limited by computational resources and faced with a trade-off between precision and number of model runs. The optimal choice is dependent on the problem; in this chapter I found that the functions and fields I chose to interpolate were well approximated by polynomials of order 6 (see below).

For this chapter, the two parameters θ_1 and θ_2 (see Section 3.2.1) are considered to be stochastic. As the prior distribution for θ_1 and θ_2 I used a uniform distribution and set the lower and upper limits of the distribution as 0.25 and 1.75 times the parameters' standard value, respectively. The standard values are taken from *Fennel et al.* (2006) and turned out to be reasonably close to the optimal (fixed) parameter set for this study (see Figure 3.3, Section 3.3.1.2). I chose the uniform distribution because of its finite support which does not permit negative parameter values, as well as yielding a simple polynomial chaos setup.[2] Finally, I selected the maximum order $k_{\text{max}} = 6$ for both parameters. As a

[2] While model uncertainty estimates might benefit from a different parameter distribution, here I rely on the polynomial interpolation aspect of polynomial chaos which is not very sensitive to changes in the distribution. Polynomial interpolation is exact at the quadrature points and a change in distribution affects the layout of the quadrature points in parameter space. Only a drastic change in the quadrature point layout can

result $(k_{\text{max}} + 1)^2 = 49$ model runs had to be performed. The 7×7 grid of quadrature points in parameter space is shown in Figure 3.3.

After performing the necessary model runs, polynomial chaos allows for the approximation of any function that is dependent on the stochastic parameters. It can therefore be used to approximate the distance function in (3.1) for the purpose of model-data comparison. Here, d takes on the role of f in equation (3.2), so $f(\mathbf{x}, t, \theta_1, \theta_2) = \mathrm{d}(t, \theta_1, \theta_2)$. As described in Section 3.2.3, the following steps are taken to approximate d. After the model is run for the parameter values of each quadrature point, the distance function is computed for each of the model runs at all time steps from 1 to n_{obs}. The expansion coefficients $a_k(t)$ specific to the distance function are then computed using equation (3.5). As the distance function d is not dependent on $\mathbf{x}$, the coefficients a_k do not depend on $\mathbf{x}$ either. Now I can use the approximation in equation (3.2) to approximate the distance function for each value of θ_1 and θ_2 in their respective ranges.

Approximating multi-dimensional fields such as the surface chlorophyll (as in the emulation experiment of Section 3.3.2.1 below) works in a similar way. The surface chlorophyll values in the topmost model layer are extracted for all model runs at all time steps. The extracted output, which is dependent on θ_1, θ_2, t and the two horizontal spatial coordinates contained in $\mathbf{x}$, is set equal to $f(\mathbf{x}, t, \theta_1, \theta_2)$. Surface chlorophyll specific coefficients $a_k(\mathbf{x}, t)$ are computed which are, like the surface chlorophyll field, dependent on the spatial coordinates $\mathbf{x}$. No recomputation of the polynomials $\phi_k(\theta)$ is necessary to obtain approximate surface chlorophyll values from equation (3.2).

3.3 Results

I hypothesized that temporal changes in plankton species composition manifest as shifts in the parameters values of the biological model. In other words, I expect that parameter values that shift in time and space will better explain the observations. The polynomial chaos expansion allows one to obtain approximations of model output for any parameter value within prescribed bounds. This property allows one to find optimal parameter values with only a limited number of computationally costly model runs. Specifically, I employed the polynomial chaos expansion to approximate the distance between observed

cause a strong effect on the polynomial interpolation but such a change would need to be caused by an equally drastic change in the parameter distribution, e.g. a strong shift in the range of the uniform distributions.

and simulated surface chlorophyll. First, I minimized the distance for the entire data set to obtain global optimal parameters independent of time and space (referred to as *optimal fixed* parameters in the following). Then I minimized the chlorophyll distance for single (daily) observations individually and for different model regions to identify temporal and spatial variations in the optimal parameter values. Both optimizations are based on the same set of 49 model runs and further require only the computationally much less demanding evaluations of the polynomial chaos-based interpolation.

3.3.1 Interpolating the Model-Data Distance Function and Parameter Estimation

3.3.1.1 Smoothing the Distance Function

I obtained estimates of optimal parameter values by interpolating and minimizing the time-dependent distance function d in (3.1). This function appears to be very noisy and varies considerably from one day to the next (Figure 3.2), not necessarily due to bad model output on days with large values of d, but because of a large number of missing values (Figure 3.2).

In order to diminish the influence of outliers in my analysis and to create a more robust distance function, I used a low-pass filter in the form of a Gaussian window to smooth d. From here on, I use the term *smoothing intensity* to describe the amount of smoothing that was applied to the distance function. The smoothing intensity is a positive integer value which increases with the amount of smoothing. More precisely, twice the smoothing intensity plus 1 is the width of the Gaussian smoothing window in days (I only use window widths that are odd), i.e. a smoothing intensity of 0 refers to a window width of $2\times0+1 = 1$ and therefore no smoothing, while a smoothing intensity of 10 refers to a window width of $2 \times 10 + 1 = 21$. Examples of the smoothed distance function and the corresponding Gaussian windows are shown in Figure 3.2. For simplicity, I do not remove any of the data outliers from my analysis, eliminating the need to create an objective criterion for their removal.

The objective of smoothing the distance function is to minimize the impact of outliers, reduce overfitting and to improve the parameter optimization.

3.3.1.2 Optimal Fixed Parameters

Typical parameter optimization studies assume fixed parameter values, and the optimized parameters are determined by minimizing the model-data discrepancy over the full set of available observations. I can do the same using the polynomial chaos expansion: To obtain estimates of the optimal fixed parameter values with respect to the distance function $\mathrm{d}(t, \theta_1, \theta_2)$ in (3.1), I eliminated the time dependence of d by computing its average in time. I then used the polynomial chaos expansion to approximate the resulting average distance function in parameter space as detailed in Section 3.2.4. The resulting distance function is smooth and exhibits a clearly defined global minimum close to the center of the domain defined by the ranges of θ_1 and θ_2 (Figure 3.3). Because the average distance function changes more along the θ_1-axis than in the direction orthogonal to it, I can deduce that model chlorophyll is more sensitive to relative changes in the value of the maximum chlorophyll to carbon ratio (θ_1) than the zooplankton grazing parameter (θ_2).

One can think of the inverse of the distance function as likelihood of the underlying parameter values (an idea that will be revisited in the following chapter). The parameter values that minimize the distance, maximize the likelihood. At this point only the maximum likelihood estimates are considered. However, the emulator allows for the approximation of the full likelihood function. I will take advantage of this for parameter estimation in Section 3.3.4.2.

In order to gauge the quality of the polynomial interpolation of the distance function, I performed a number of analysis model runs along a slice through the parameter domain (green line in Figure 3.3(a)). A comparison of the approximated distances (light green line in Figure 3.3(b)) with the exact distances obtained for the analysis runs (red symbols in Figure 3.3(b)) reveals that the average distance function is generally well approximated by the interpolation for k_{max} with only some edge effects typical of polynomial approximations. This leads me to conclude that the position of the global minimum of the average distance function in Figure 3.3(a) represents a good approximation of the optimal parameter values with respect to the full data set. In the following I will refer to the parameter pair that minimizes the interpolated average distance function as the optimal fixed parameters.

The analysis model runs can also help assess the convergence of the polynomial chaos approximation. I chose $k_{\text{max}} = 6$ for the approximation in equation (3.2). The effects of truncating the sum at lower orders (smaller values for k_{max}) are shown in Figure 3.3(b). The results of the analysis model runs remain fairly well approximated for $k_{\text{max}} \geq 4$, but below that, the approximation becomes considerably worse. Interestingly, the position of the minimum changes relatively little with the addition of higher order polynomials.

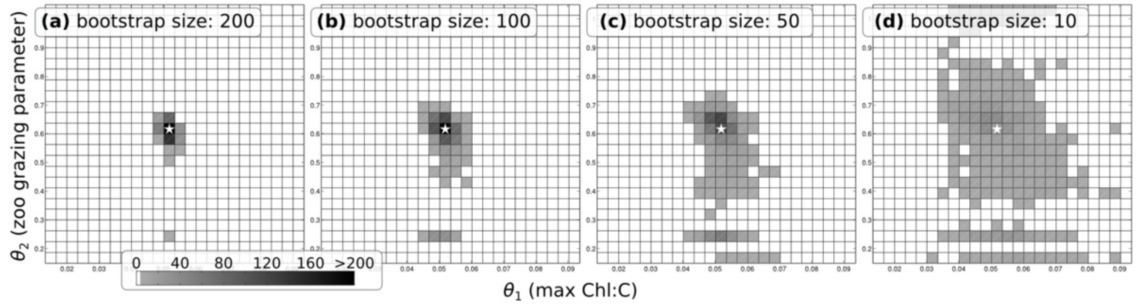


Figure 3.4: Two-dimensional histogram of the position of the global minimum of the time-averaged distance function in parameter space (compare Figure 3.2) for the bootstrapping experiment with 1000 bootstraps described in Section 3.3.1.2. With decreasing bootstrap size, the location of the minimum becomes more variable, especially along the axis corresponding to the parameter θ_2. The white star in the center of each image is a reference point.

For the purposes of this chapter, the position of the minimum of the average distance function is approximated well and choosing a higher $k_{\max}$ at the cost of additional model runs appears unnecessary.

For a different data set or a subset of the data, the average distance function and the position of its minimum is likely to change. It is desirable to gain an understanding of the uncertainty in the position of the global minimum given in Figure 3.3. For this purpose, I performed a bootstrapping experiment: I generated subsets of the observations (the bootstraps) by randomly selecting a fixed number of satellite images from the 350 images that make up the complete observational data set. For each bootstrap, I calculate the global minimum of the respective time-averaged distance function. For the relatively large bootstrap size of 200 images, drawn without replacement, I see a tight clustering of minima around the full data minimum (Figure 3.4(a)). With a decrease in bootstrap size, the range becomes greater, especially along the θ_2 axis. At the small bootstrap size of 10, the minimum positions are distributed all along the selected range of θ_2 (Figure 3.4(d)).

It is apparent that the optimal fixed parameter set is very much dependent on the subset of data used in the optimization exercise and can vary considerably based on its choice. In the following sections, I show that this dependence is mainly due to an underlying time-dependence of the optimal parameters and not primarily due to the noise contained in the data set.

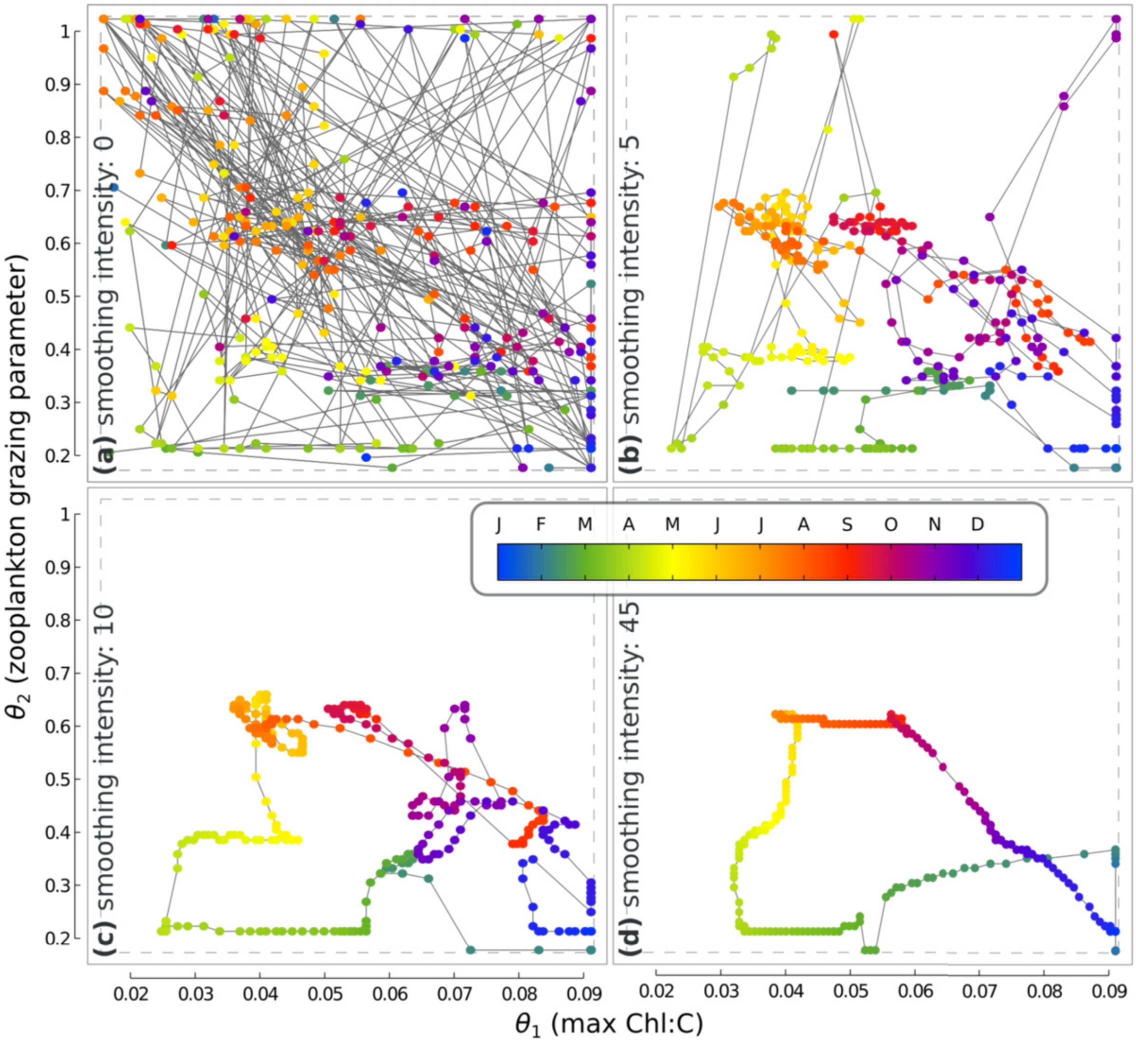


Figure 3.5: The parameter paths obtained by minimizing the time-dependent distance function in each time step for 4 different smoothing intensities. The smoothing intensities (0, 5 and 10) in panels (a) to (c) correspond to those shown in Figure 3.2.

3.3.1.3 Time-Varying Parameters

Time-dependence of the optimal values of the physiological parameters θ_1 and θ_2 suggests that there is a signal in the observations that the model cannot account for if the parameter values are fixed. To uncover time-dependence, I return to the polynomial chaos approximation of the distance function. In the previous section I used it to obtain a set of optimal fixed parameters for the entire data set by minimizing the average distance function. Using a very similar procedure, I can approximate the distance function for each daily observation to obtain a set of optimal parameters for each day. In other words, I used the polynomial chaos expansion to interpolate $\mathrm{d}(t, \theta_1, \theta_2)$ in parameter space for $t = 1, \ldots, n_{\text{obs}}$ and determine the global minimum of the function for each t. I performed this procedure for the unsmoothed version of d as well as for versions that have been smoothed at different intensities as described in Section 3.3.1.1. Then I arranged the resulting parameter values for each smoothing intensity into a parameter path in time, as shown in Figure 3.5. The path corresponding to the unsmoothed distance function appears very jagged, dominated by high frequency variation and with little structure; as the temporal smoothing increases the paths become more structured and a loop emerges. It is important to note that the optimal parameter values obtained this way correspond to the model simulations with fixed parameters. In other words, the optimal parameter values at any date, say 15 August, correspond to a model simulation that has been running with these fixed parameters until 15 August. It is dependent on the model if the parameter paths obtained from these fixed parameter simulations are suitable as time-dependent parameters of the model and translate into improved state estimates. For the model used here, this is tested in experiments in Section 3.3.3, and the issue is further discussed in Section 3.4.

The structure of the parameter paths can be interpreted in a straightforward way. With no smoothing, the procedure picks the optimal parameter set to match one satellite image alone, including the noise contained within the image. The distance to the previous or following image is not considered. As the distance function is noisy (see Figure 3.2), I expect a high amount of noise in the daily optimal parameter values as well. The high frequency variations in the daily optimal parameters are therefore likely local fits to the noisy data. However, Figure 3.5 also shows clear evidence of a low frequency parameter change visible at higher smoothing intensities. This low frequency signal reveals that there is a time-dependence of the optimal parameter values that cannot be explained by the noise in the observations, indicating that the fit between model and observations can be improved by allowing parameters to follow the low frequency signal using cross-validation.

These results also suggest that there is an optimal smoothing intensity, strong enough to filter out the effects of the noise contained in the data, yet not too strong to also remove the low frequency signal I am interested in. In the following section, I show how the chlorophyll output of the model can be improved by using the low frequency parameter paths. Based on a comparison with the observations I also determine the optimal smoothing parameter that best isolates the low frequency signal.

3.3.2 Emulating Surface Chlorophyll

3.3.2.1 Polynomial Chaos-Based Emulation

In the previous section, I described how time-dependent parameter paths can be obtained from the interpolation of the likewise time-dependent distance function. Here I utilize these paths to obtain improved model estimates of surface chlorophyll fields. I use the polynomial chaos expansion as an emulator, i.e. a system that allows me to obtain estimates of the state of the ocean for a parameter combination I did not perform a model run for. In this chapter, I emulate the full surface chlorophyll field using the polynomial chaos based-interpolation.

As described in Section 3.2.4, the polynomial chaos expansion can be used to interpolate virtually any model output in parameter space, including the time-dependent chlorophyll concentrations in the surface layer of the model. This feature allows one to efficiently interpolate the chlorophyll values along the parameter paths. I obtained daily pairs of parameter values from one of the time-dependent parameter paths (see Figure 3.5). Then, with the help of the polynomial chaos expansion, I estimate the surface chlorophyll fields that correspond to the daily parameter values. This procedure results in an emulated time-dependent surface chlorophyll field, which is dependent on the smoothing intensity that underlies the chosen parameter path. Note that one can use the same procedure to obtain estimates of depth-resolved chlorophyll fields or other biological properties along the parameter paths.

I then compared the interpolated chlorophyll fields to the observations as in previous sections, using the same distance measure d in (3.1) but replacing chlorophyll model output with the interpolated model chlorophyll fields. This way, I obtain a distance value for each day which, averaged in time, results in an average distance value. I computed average

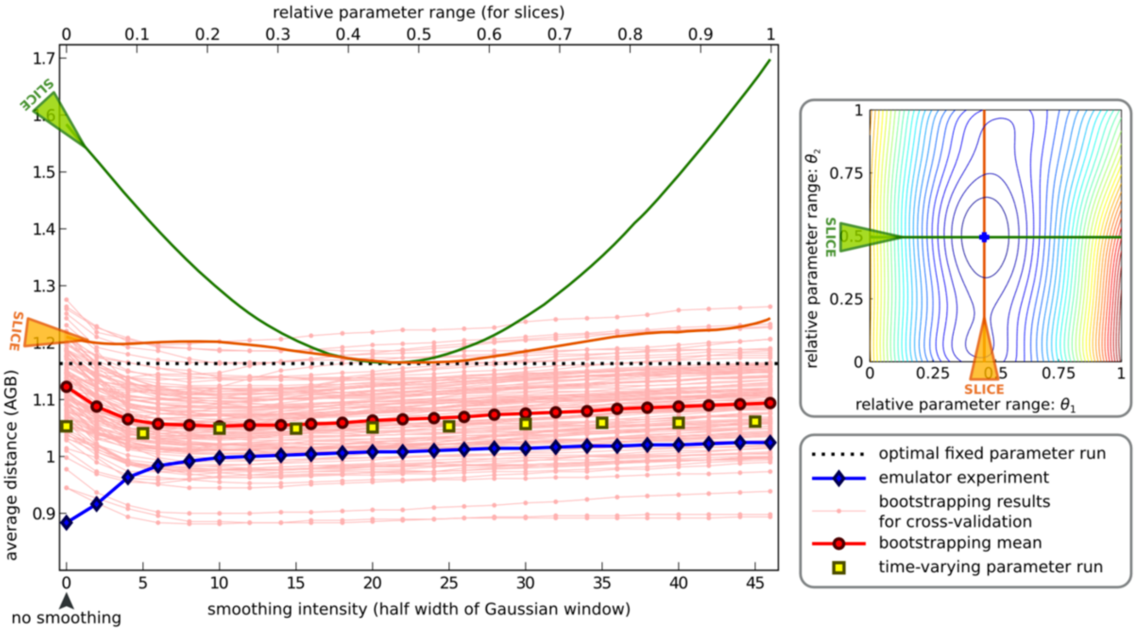


Figure 3.6: The average distance values of various experiments: The emulation experiment (Section 3.3.2.1), the bootstrapping results for the cross-validation experiment (Section 3.3.2.2) and the time-varying parameter runs (Section 3.3.3), each dependent on the smoothing intensity. They are compared to the average distance value obtained by the optimal fixed parameter run (Section 3.3.1.2) which is independent of the smoothing intensity. For scale, two slices of the average distance function for fixed parameters (see Figure 3.2) are displayed in the same plot.

distance values for different smoothing intensities (blue diamonds in Figure 3.6).

The resulting average distance values based on the emulation experiment are smallest for the parameter path without smoothing (Figure 3.3(a)), and increase with more smoothing. They are directly comparable to the average distance value of the optimal fixed parameter model run from Section 3.3.1.2. For smoothing intensities up to 45 all average distances are well below the values of the optimal fixed parameter run (indicated by the dashed line in Figure 3.6). In other words, the emulated chlorophyll fields are considerably better than those of any model run with fixed parameters, which was to be expected. The fact that the lowest distance is associated with the no-smoothing path, indicates that at least part of the improvement is due to overfitting the data. At low smoothing intensities the emulated values fit even outlying values and noise very well, completely disregarding the model dynamics.

In the following section I perform a cross-validation to address this issue and determine for which smoothing intensities overfitting is not a concern. The cross-validation also allows me to identify the optimal level of smoothing.

3.3.2.2 Choosing the Optimal Smoothing Parameter in a Cross-Validation Experiment

The jagged nature of the parameter paths at low smoothing intensities (Figure 3.5(a)) indicates overfitting of the model to the observations. Cross-validation experiments provides a technique to distinguish overfitting from real improvement in model performance. I follow the typical approach where the observational data set is partitioned into two parts, the training set and the validation set. The training set is only used to optimize the models parameters, the quality of the model output is then assessed by a comparison with the validation set. Overfitting the training set will not lead to a better model performance with respect to the validation set.

I performed multiple cross-validation experiments in a bootstrap fashion. In each experiment, the observations were split into training and validation set in the following way. The training set contains the first and last satellite image as well as a number of randomly selected images in between; the validation set consists of the remaining images. I then performed 25 cross-validation experiments for each of five training set sizes (175, 150, 125, 100 and 75) ranging from half of the observational data set to roughly one fifth. In each experiment, I determined the parameter path according to the procedure described in Section 3.3.1.3, but only using the training data set. This way I obtained optimal parameter values corresponding to the time steps of the training data. I determined the quality of these parameter values with respect to the validation data set in 3 steps: (1) I linearly interpolated the parameter values corresponding to the training set dates in time to obtain the parameter values for the validation set dates. (2) I used the freshly obtained parameter values to interpolate the surface chlorophyll field in parameter space, yielding a surface chlorophyll field for each validation set date. (3) With the standard distance measure, I computed the distances of the surface chlorophyll fields to the validation data and calculated the average distance.

The average distance values obtained through the above procedure are shown as red dots in Figure 3.6, and exhibit a clear difference compared to the values of the emulation experiment without cross-validation (Figure 3.6, blue diamonds). First of all, the cross-validation distance values are generally higher than those of the emulation experiment. This is to be expected from a cross-validation experiment which uses two separate data sets for optimizing parameters and assessing the fit. More important is another difference: While the setup without cross-validation has the lowest distance at a smoothing intensity of 0 and then increases steadily, the cross-validation mean has a minimum at a smoothing intensity of 10, corresponding to a smoothing window width of 21 days. The minimum is relatively flat towards higher smoothing intensities but shows a sharper incline for intensities lower than 5. This property is strong evidence for the presence of overfitting at low smoothing intensities. For no or little smoothing the jagged parameter path describes the noise in the observations and the parameter values do not generalize well to the validation data set in the cross-validation. As the smoothing is increased, overfitting becomes less of a problem and disappears. When smoothing is increased even further, useful information in the observations is filtered out so that average distances increase again, albeit at a slow rate. I therefore consider a smoothing intensity of 10, the position of the minimum of the cross-validation mean curve, the optimal smoothing intensity for my emulation experiment, and use it as the standard smoothing intensity for the emulation experiment in the following section.

In this section and the previous one, I have shown that the time-dependent parameter paths in combination with state interpolation can be used as a emulation tool that produces state estimates which are considerably better than those of any fixed parameter model run.

The improvement is not due to fitting noise in the data, as the smoothing intensity can be adjusted to avoid the problem of overfitting; it is due to the presence of an underlying time-dependence or seasonal cycle in the parameters. In the following, I assess the utility of the parameter paths for obtaining time-dependent parameter values for the biological model.

3.3.3 Model Runs with Time-dependent Biological Parameters

In addition to obtaining improved estimates of chlorophyll by means of a polynomial chaos interpolation, the parameter paths can also be used in a more straightforward way. One can perform biological model runs with time-varying values of θ_1 and θ_2 by plugging parameter paths directly into the model. The values of the two parameters are taken from a specific parameter path and so the results are again dependent on the smoothing intensities used to obtain the path.

To implement time-varying parameters in the model I extended the parameter paths which are defined only for the discrete time steps $t = 1, \ldots, n_{\text{obs}}$, to the interval $[1, n_{\text{obs}}]$ by linearly interpolating the paths in time. In the numerical model this was implemented by incorporating a simple lookup table for the parameter values at $t = 1, \ldots, n_{\text{obs}}$. At each model time step the model looks up the values of θ_1 and θ_2 that correspond to the two closest points in time and performs the time interpolation. By using different lookup tables, one can perform model runs for different parameter paths or smoothing intensities. I set the initial values of θ_1 and θ_2 to the first value of the parameter path and ran the model, keeping all other settings unchanged. Again, I computed the average distance values for the time-varying parameter runs (Figure 3.6, yellow squares).

For the time-dependent parameter runs, the lowest average distance is achieved at a smoothing intensity of 5 (Figure 3.6, yellow squares), although there appears to be no strong dependence on the smoothing intensity as all distance values are very closely grouped. Generally, the time-dependent parameter results are in between those corresponding to the emulation experiment described in Section 3.3.2.1 (Figure 3.6, blue diamonds) and the optimal fixed model run results (Figure 3.6, dashed black line). One would expect a degraded performance of the time-varying runs in comparison to the emulated results given the memory of the model with time-dependent parameters, an effect that I will reconsider in the discussion in Section 3.4. However, the improvement in average distance in comparison to the optimal fixed parameter run is still large. In comparison with the

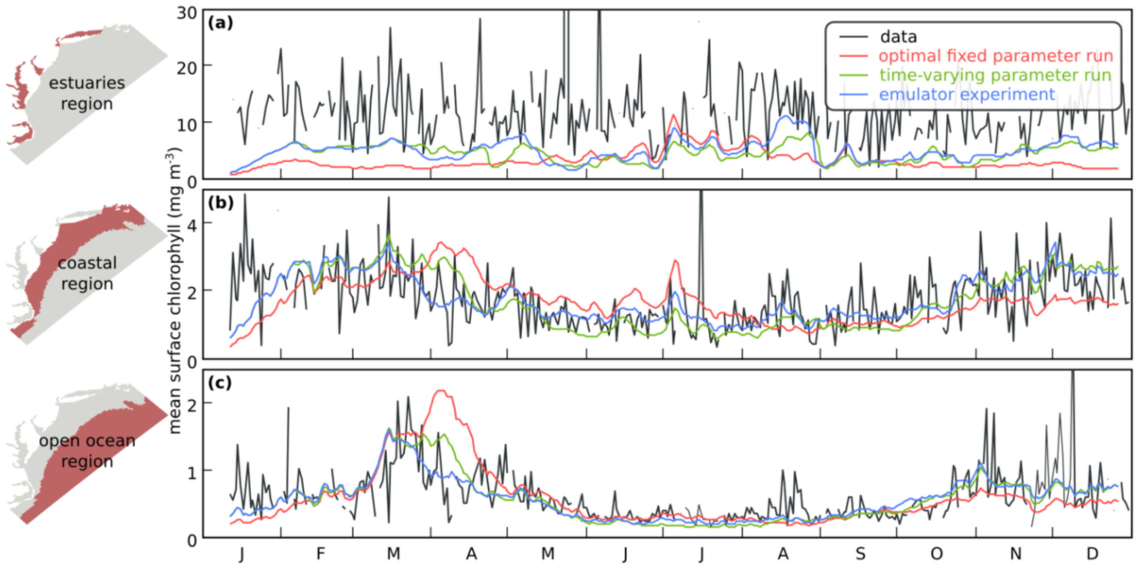


Figure 3.7: The development of the average concentration of surface chlorophyll for the model, the optimal fixed parameter run, the time-varying parameter run (with a smoothing intensity of 5, corresponding to the best average distance result in Figure 3.6) and the emulation experiment (with a smoothing intensity of 10, corresponding to the lowest average distance in the cross-validation experiment in Figure 3.6). The analysis is divided into 3 model regions which are displayed in the left panel. The corresponding absolute residuals are shown in Figure 3.8.

optimal fixed values of θ_1 and θ_2, the model creates considerably better chlorophyll output if I allow the values to change in time. As the model reacts relatively slowly to shifts in parameter values and the average distance values of the different time-varying runs are very similar I can conclude that these runs do not overfit the data. To further assess the improvement I performed a follow up comparison of the estimated chlorophyll values using the 3 estimators: the optimal fixed parameter run, the time-dependent parameter run and the emulation experiment, in the following section.

3.3.4 Temporal and Spatial Analysis

3.3.4.1 Spatial Comparison of Chlorophyll Estimates from Model and Emulator

In order to asses how the differences in average distance values for the 3 runs, the optimal fixed parameter run (Section 3.3.1.2), the time-dependent parameter run (Section 3.3.3) and the polynomial chaos-based emulation experiment (Section 3.3.2.1), translate into differences in surface chlorophyll I calculated the regional chlorophyll averages for 3 regions of the model domain, the estuaries, the coastal and the open ocean region (Figure 3.7). In the estuaries, all model estimates of chlorophyll underestimate the observations (Figure 3.7(a)).

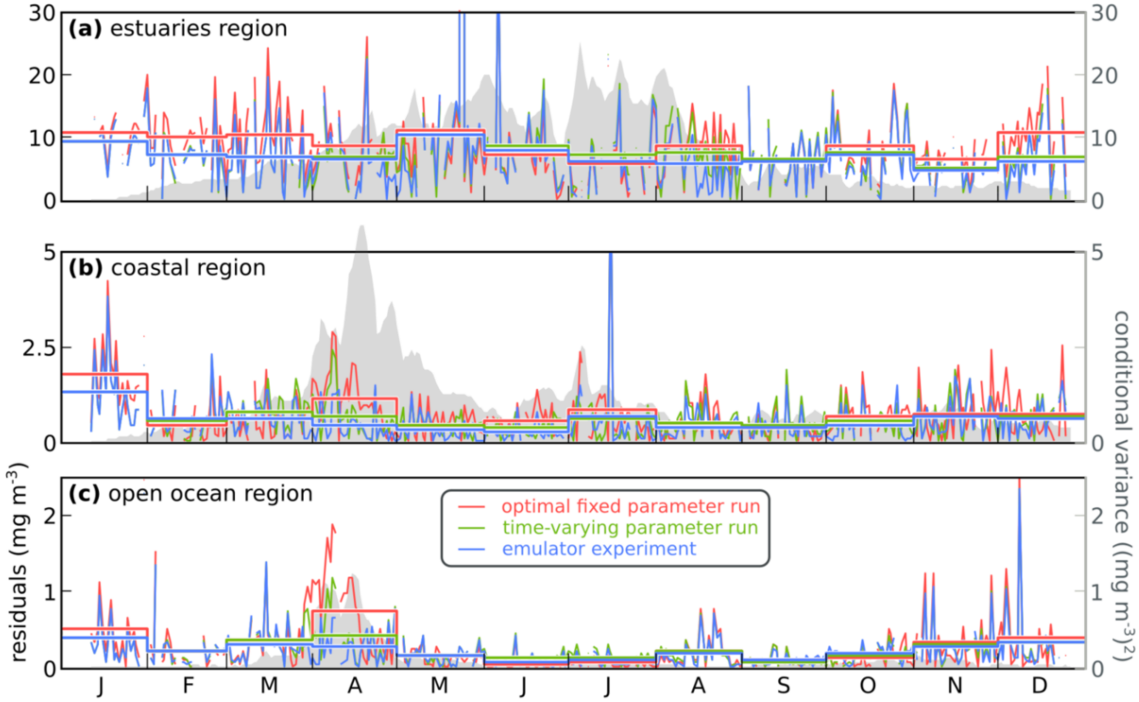


Figure 3.8: The absolute residuals of the surface chlorophyll content shown in Figure 3.7; monthly averages of the absolute residuals are displayed as thick lines. The grey area in the background is the variance of the surface chlorophyll content based on equation (3.6).

This result is not unexpected, as the relatively coarse resolution model cannot adequately represent estuarine dynamics. Additionally, satellite chlorophyll estimates might be biased high due to high levels of coloured dissolved organic matter in the water which is known to enhance the chlorophyll signal in satellite images (*Mannino et al.*, 2008).

The model estimates agree better with the observations in the other two regions, the coastal region and the open ocean. In both regions it is also apparent that the time-varying parameter model run and the emulated state estimates show improvement over the optimal fixed parameter run. A look at the deviations from the data, shown in Figure 3.8, reveals that surface chlorophyll estimates are indeed most accurate for the emulated state estimates, followed by the time-varying parameter model run and the fixed parameter model run. Improvement is most evident in April, during the spring bloom. In a few instances, the fixed parameter model produces the lowest absolute residuals in some regions of the model. These are however offset by higher residuals in other regions (compare, e.g., the June residuals in Figure 3.8 across all 3 regions). This demonstrates that there is no uniform improvement across the entire model domain, instead the improvement achieved by time-varying parameters depends on both time and space.

Generally, improvement is more likely where the parameter variation induces the greatest variance into the surface chlorophyll state. This observation follows from a comparison of the absolute residuals with the variance (see equation (3.6); shown as grey area in Figure 3.8). Where the variance is high, a change in the parameter values has a large effect on the surface chlorophyll concentration. This, in turn, allows for more effective adjustments of the chlorophyll concentration by means of changing θ_1 and θ_2.

3.3.4.2 Spatial Differences in Optimal Parameter Values

Based on the optimal smoothing intensity found in Section 3.3.2.2, I now re-evaluate the development of optimal parameter values in time and examine the uncertainty in the model state. Instead of using the minimum as a point estimate for an optimal parameter value, I am interested in a *region* of good parameter values. These values are "good" in the sense that they are associated with low (but not necessarily minimal) distance values and therefore high likelihood values. To determine good parameter values, I performed the following steps: (1) For each day with available data, I interpolated the corresponding distance function in parameter space using the polynomial chaos expansion. (2) For each of the distance functions, I determined the region in parameter space that makes up 20% of its lowest values (20% of the highest likelihood). (3) Finally, I computed the frequency with which a given pair of parameter values is contained within the 20% region. I expect that a good pair of parameter values is contained frequently in the 20% region of low distance. The frequency of occurrence in this region, obtained for all parameter values, therefore provides an estimate of the distribution of good parameter values which can be visualized easily.

Estimates of the parameter distribution for each season (Figure 3.9(a)), obtained by the procedure described above, correspond well to the parameter paths in Figure 3.5, yet the distribution additionally reveals features hidden in the point estimates. For example during the spring (AMJ; corresponding to April, May and June) there appears to be very little sensitivity to changes in θ_2, the zooplankton grazing parameter, and good parameter values are distributed all along the θ_2 axis. In summer (JAS; July, August, September), the distribution changes in this respect, as low values of θ_2 become less probable as good parameter values. Seasonal differences are generally apparent, strengthening my previous observations that optimal parameter values change in time.

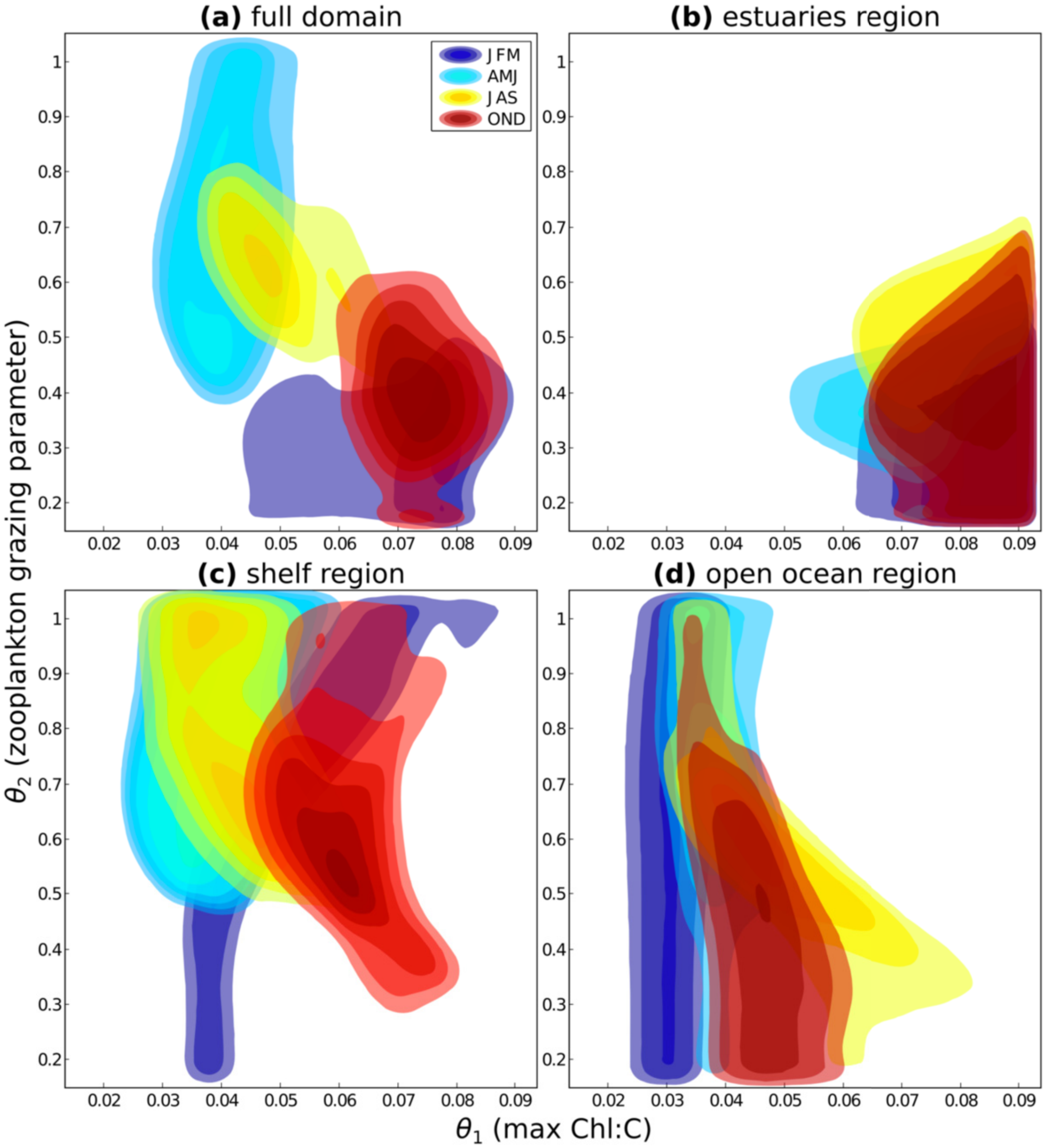


Figure 3.9: The distribution of good parameter values in parameter space depending on season, the contours mark the regions of good parameters (regions in parameter space containing consistently 20% of the lowest distance values throughout the season; compare Section 3.3.4.2). The first panel shows a contour plot of the distributions for the entire model domain, the other panels contain the distributions for the 3 regions shown in Figure 3.7.

So far, I focused mostly on the change of parameters in time, but I can also use the same methodology for an analysis of spatial differences. For a spatial analysis, I use the 3 model regions introduced in Section 3.3.4.1 and shown in Figure 3.7. All previous results were based on the distance function introduced in (3.1) which uses the full data set to compute distance values. By including observations from within one of the 3 regions only, the distance values can be recomputed and one can gain an understanding of suitable parameter values for that region. In order to detect spatial differences in good parameters I performed the distribution estimation for the 3 regions again (Figure 3.9(b,c,d)).

Differences between model regions are apparent. In the estuaries, where chlorophyll is always underestimated, good parameter combinations tend to increase chlorophyll by combining high values of the chlorophyll-to-carbon ratio with low values of the zooplankton grazing rate throughout the whole year. More temporal variation is evident in the other two regions. In the shelf region, seasonal changes are most apparent and values of the zooplankton grazing rate tend to be generally high, especially in spring and summer. This result corresponds well to the tendency of the optimal fixed parameter run to overestimate chlorophyll during those months. In the outer ocean region which exhibits the lowest chlorophyll values, the model tends to be most insensitive to changes in the zooplankton grazing parameter whereas a very narrow range of θ_1 is preferred. The low amount of chlorophyll combined with relatively little chlorophyll variability sustains only a small population of zooplankton, thus the grazing parameter of zooplankton has a low impact.

Taken together, the results for the 3 model regions account for the full domain result presented above. The distance measure that was used (AGB) has no knowledge about the regions, thus the influence of the regions on the general result is mainly determined by their size. Hence, the large coastal and open ocean regions far outweigh the influence of the small estuaries region. Due to their different parameter preferences, the fit between data and model remains relatively poor for the estuaries region (compare Figure 3.8). Despite being small, the estuaries exert a constant influence on the parameter estimation to raise chlorophyll levels. Here the polynomial chaos based interpolation shows its strength as a model analysis tool.

3.4 Discussion

In this chapter I obtained improved surface chlorophyll estimates from a biological ocean model by treating two of its parameters as stochastic. This was achieved through the approximation of the model by a low dimensional emulator, the polynomial chaos expansion. Using the polynomial chaos expansion in combination with a model-data distance function I found that the probability densities of the two biological parameters have a clear time dependence and their expected values follow a seasonal path through parameter space (Figure 3.5).

At two points in this chapter high frequency variations were encountered; they appeared in the distance function (Figure 3.2) and the derived parameter paths (Figure 3.5). In the case of the distance function I attribute the high frequency signal to noise and missing values in the chlorophyll images. The high frequency changes in the parameter paths indicate that the same noise is overfitted by my optimization procedure. I confirmed this inference in a cross-validation experiment (Section 3.3.2.2), where I observed a strong increase in the average distance value for low smoothing intensities while the best results were achieved at medium smoothing intensities (Figure 3.6). This result is evidence that the improvement of my time-varying parameter state estimates is based on an actual signal in the observations that is not captured in the fixed parameter run.

By treating only two biological parameters as stochastic and by adjusting them to fit the observations, I do not account for the fact that model-data discrepancies are also caused by other sources of model error, such as the other biological parameters, parameters of the underlying physical model, physical forcing, boundary and initial conditions as well as the functional form of the equations themselves. For example, the selection of the maximum chlorophyll-to-carbon ratio (θ_1) as a stochastic parameter and its optimization may adjust for errors in the phytoplankton growth rate and errors in the model's nutrient supply. A fully Bayesian approach, which would incorporate all sources of model uncertainty, is computationally infeasible. I chose to focus this chapter on one obvious shortcoming of the model, the representation of phytoplankton and zooplankton as homogeneous groups. Within this much more limited scope, I selected the two parameters that have the strongest influence on the model's chlorophyll concentration. Here, my motivation is simply that the most sensitive parameters will likely be identifiable using chlorophyll data and yield the biggest improvement in chlorophyll estimates.

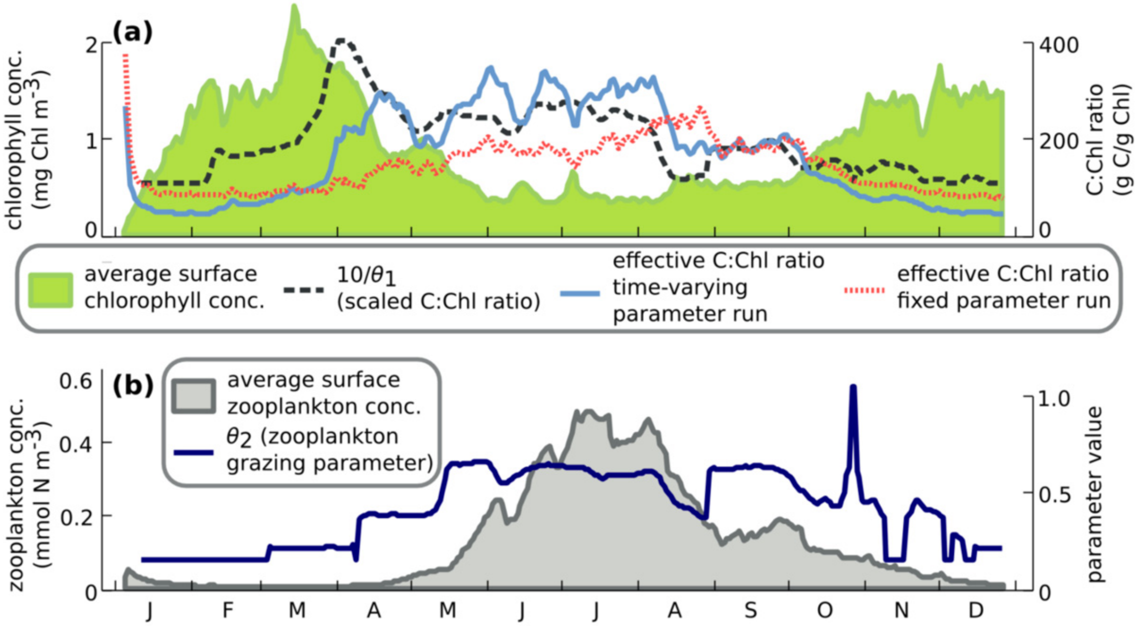


Figure 3.10: The development of the time-varying values of θ_1 and θ_2 (smoothing intensity of 10, compare Figure 3.5) in relation to the development of the surface chlorophyll content. In addition to θ_1, panel (a) also shows the resulting carbon to chlorophyll (C:Chl) ratio in the surface time-varying parameter run and the corresponding C:Chl ratio for the optimal fixed parameter model run. For better comparison, θ_1 is transformed to $\frac{1}{\theta_1}$ which is also a carbon to chlorophyll ratio and then scaled by a factor of 10.

Although I have no detailed information on the phytoplankton species succession and seasonal changes in grazing rate, I can attempt a qualitative comparison of the development of θ_1 and θ_2 with typical seasonal changes in the plankton composition of the Middle Atlantic Bight. In the model run with time-varying parameters, there is a positive correlation between the inverse of the maximum chlorophyll-to-carbon ratio ($\frac{1}{\theta_1}$) and the achieved phytoplankton carbon-to-chlorophyll ratio (C:Chl) in the surface (Figure 3.10(a)). In comparison to the model with optimal fixed parameters, the time-varying parameters lead to an increase in C:Chl in the summer months following the phytoplankton spring bloom. In the Middle Atlantic Bight dinoflagellates typically dominate the phytoplankton community in the shelf region during summer (*Marra et al.*, 1990) while diatoms are the dominant phytoplankton group during the spring bloom (*Barlow et al.*, 1993). Due to a significantly lower C:Chl in diatoms in comparison to dinoflagellates (*Chan*, 1980), I would expect a lower C:Chl during the spring bloom and a higher C:Chl in summer. While the optimal fixed parameter run shows no marked increase in C:Chl as the bloom subsides, there is a notable increase in the C:Chl induced by the time-varying parameters, consistent with the expected C:Chl development (Figure 3.10(a)). This improved correlation does not imply causation, as I have pointed out in the previous paragraph, yet it is consistent with the hypothesis that variations in C:Chl are significantly affected by shifts in the phytoplankton composition.

Evaluating the development of θ_2, the zooplankton grazing parameter, is more difficult. In the model, the zooplankton maximum occurs in the summer, is preceded by a notable increase in θ_2 in April (Figure 3.10(b)) and remains high for several months until November. This pattern may reflect a correction of the seasonal cycle of zooplankton. The increase of θ_2 in April enhances zooplankton grazing and hints that the effect of grazing in the model is too low at that time of the year. *Kane* (2005) found that the zooplankton species *Calanus finmarchicus*, an important part of the zooplankton population, show a sharp increase in abundance in early spring and suggest that it is a consequence of an import of zooplankton into the Middle Atlantic Bight from neighbouring regions. Such a process is unaccounted for in the biological model and could explain the development of θ_2 in spring. Lack of import causes an underestimation of zooplankton abundance and grazing in the model, which is counteracted by an increased zooplankton grazing parameter. It should be noted, however, that it may be difficult to constrain the zooplankton parameter using chlorophyll observations, given the indirect effect of changes in zooplankton grazing on chlorophyll. In addition, zooplankton dynamics are known to be highly variable from one year to the next, even under similar phytoplankton conditions (*Flagg et al.*, 1994).

Given the self-imposed restriction of optimizing two parameters, the improvement in surface chlorophyll estimates is considerable. By using the parameter paths as time-varying parameter values, more improvement can be achieved than by changing θ_2 (the zooplankton grazing parameter and one of the models most sensitive parameters) from its most disadvantageous value in the broadly selected parameter range, to its optimum value (Figure 3.3). The results of the emulation experiment tend to be better than those of the biological model simulation with time-varying parameters; the main reason for this is that the emulation experiment is not bound by the model dynamics and changes in parameters become effective immediately. In contrast, the time-varying parameter model run has a memory of accumulated (or lost) chlorophyll and a change in parameter value needs some time to translate into a changed surface state. I expect that the level of improvement to be gained from time-varying parameters, will in general depend on the model's memory, where properties with fast response will be more prone to improvements.

I varied only two of the biological parameters and decided to keep the general setup simple, e.g. by using the entire data set without excluding outliers. The distance function I interpolated appears to be smooth and well approximated by the polynomial interpolation (Figure 3.3). Consequently, I can still expect good results for fewer quadrature points in parameter space, which has the benefit of decreasing the number of necessary model runs. Yet even after a reduction of quadrature points it would be computationally expensive to extend the analysis to more than a few parameters. Other emulator approaches can sample parameter space in a more efficient, non-grid based manner (e.g. latin hypercube sampling introduced by *McKay et al.* (1979) or a free selection of parameter values as in *Hooten et al.* (2011)) and may be better suited for parameter estimation in higher dimensional spaces. One advantage of the polynomial chaos technique is that it offers a straightforward way to obtain model uncertainty estimates (see Section 3.2.3), which do not require an additional analysis step. In contrast to other emulators that do not utilize basis functions, there is no need to run Monte Carlo-based sampling techniques within the emulator framework to obtain approximates of model uncertainty (integrals of interest can be evaluated directly with the help of the polynomial basis functions).

I focused this chapter on one specific data type, satellite images of chlorophyll, in conjunction with one specific model-data distance measure, the AGB. The approach I took to estimate optimal parameters and further obtain improved state estimates is very flexible and allows for the use of other model-data distance measures (such as RMSE), other data types (such as in-situ measurements) and combinations of different observations. Any model-data distance measure suitable for the comparison of the data type of choice or a (weighted) sum of multiple such distance measures would have to be substituted for the distance function d in (3.1). The polynomial chaos expansion can then operate on the new distance values without any further changes. In fact, the use of one or more new data sets or new distance measures does not require new model simulations.

A great advantage of the polynomial chaos expansion is the amount of postprocessing and analysis options. Once the necessary model simulations have been performed (in this case 49 runs) various different analyses, from distance function interpolation, to spatial analyses and chlorophyll surface state interpolations, can be performed without any further model simulations. In addition to direct estimates of the model output, the polynomial chaos expansion also provides estimates of the variance (equation (3.6)) of the output. While it is not a measure for the full model error, knowledge of the variance can be useful for analyzing the model output, for example, to gauge the impact of the parameter variation on a specific model region or time. In my analysis, the variance (Figure 3.8) gives a good indication where in space and time model improvement is possible by means of parameter optimization. Given these advantages, I consider the polynomial chaos expansion a useful tool for model analysis and the introduction of uncertainty into biological models.

This chapter offers some insights into general parameter optimization issues. The average model-data distance function is well behaved and contains a clearly defined global minimum (Figure 3.3), which even simple parameter optimization techniques will find easily. Yet its smoothness hides the fact that the optimal parameters for individual observations are widely scattered in parameter space (Figure 3.5). Part of the reason for the wide spread of optimal parameter values is the strong underlying time dependence. Was I to optimize the model with fixed parameters using only satellite data from spring months, I would get significantly different results than by using fall data (Figure 3.9). By optimizing the model with a full year's worth of observations the fixed parameter values fall somewhere in between the optimal seasonal values. For this chapter, only one year of daily satellite observations was used. One of my next steps will be to analyze if the parameter paths generalize well for other years.

3.5 Conclusions

The model-data fit of a typical biological ocean model can be greatly improved by allowing biological parameters to vary in time. I obtained the parameter values of two biological parameters by minimizing a time-dependent distance function using an emulator-based approximation. State estimates that are based on the time varying parameters fit observations much better than those gained from the optimal fixed parameter run. This improvement is not due to overfitting the data, instead there is a low frequency variation present in the parameter values: The two biological parameters analyzed here appear to follow a seasonal cycle in parameter space in the mean. The development of at least one of the parameters matches patterns observed in plankton dynamics in the Middle Atlantic Bight.

Beside temporal differences, I also detect spatial differences of optimal parameter values for selected model regions. The estuaries and coastal and open ocean regions in the model domain show clear preferences for distinct parameter values. The polynomial chaos expansion can help identify spatial differences, detect model regions with a generally bad fit to the data and assess their influence on optimal parameter values.

The polynomial chaos expansion proved to be a versatile tool for the optimization and analysis of the biological model. While computational cost limits the number of parameters one can analyse jointly to just a few, large gains can be achieved by analysing only two parameters to which the model is sensitive. The number of postprocessing options is great: model uncertainty estimates can be obtained directly, and multiple parameter estimations with different data sets can be performed efficiently without the requirement for any additional model runs.

CHAPTER 4

PARTICLE FILTER-BASED DATA ASSIMILATION FOR A 3-DIMENSIONAL BIOLOGICAL OCEAN MODEL AND SATELLITE OBSERVATIONS

4.1 Introduction

Numerical ocean models are growing in their capabilities and their significance for ocean research and prediction. At the same time, an ever larger number of observing platforms, from underwater observatories to satellites provide an unprecedented wealth of ocean observations. Data assimilation procedures, which combine models and observations, represent the principal means by which a large number of observations can be used to improve model estimates and forecast abilities, and quantify uncertainty. Beside emulator-based methods which were introduced in the previous chapter, two major categories of data assimilation methods are: (i) variational methods, such as 4DVAR (*Bennett*, 2002), and (ii) ensemble methods, such as the ensemble Kalman filter (EnKF) (*Evensen*, 2009). Ensemble, or sample-based, approaches have become popular in recent years due to their relative ease of implementation, and their straightforward treatment of model error. They treat the data assimilation problem from a probabilistic perspective, viewing dynamical prediction as a Markov process, and the blending of model predictions with observations from a Bayesian perspective (*Wikle and Berliner*, 2007).

Particle filters represent a very general class of ensemble-based statistical data assimilation techniques that offer complete solutions to nonlinear and non-Gaussian data assimilation problems (*Ristic et al.*, 2004). Unlike the EnKF, there are no restrictive assumptions about the probability distributions (the EnKF assumes that the prediction and filtering distributions are normal, and that the likelihood function is linear and normal). When these assumptions are violated the EnKF solutions are sub-optimal (*van Leeuwen*, 2010), which has been demonstrated for nonlinear systems (*Dowd*, 2007). Hence, the generality of the particle filter makes it attractive for application to the complex, highly nonlinear models encountered in oceanography and meteorology. Yet, particle filters have been applied infrequently due to practical problems in the ensemble representation in high dimensional applications (*van Leeuwen*, 2009, see also discussion in Section 4.5). Here, I present a particle filter that focuses on parameter and chlorophyll state estimation for a 3-dimensional biological ocean model, a large-scale system with a high-dimensional state space. In this application a number of modifications are introduced that improve the particle filter's robustness and allow for effective data assimilation.

As a basis, I use sequential importance resampling (SIR; *Gordon et al.*, 1993; *Kitagawa*, 1996), the most standard particle filter algorithm. Like all particle filters, SIR is a sequential or recursive technique which assimilates observations in sequence, a setup that is suitable for online estimation, i.e. forward operation and prediction. For the purpose of state estimation, particle filters use an ensemble of model simulations, run in parallel, that allow for the approximation and propagation of model uncertainty. While applications of other ensemble techniques, especially the EnKF, are numerous in oceanography (*Ciavatta et al.*, 2011; *Hu et al.*, 2012), most SIR particle filters have only been applied in the context of relatively simple models (*Dowd*, 2007; *Mattern et al.*, 2010a) or using synthetic (typically model-generated) observations (*Annan and Hargreaves*, 2010) which do not contain outliers or observation errors. In this chapter, SIR is applied to a realistic 3-dimensional biological ocean model of the Middle Atlantic Bight in the North Atlantic (*Hu et al.*, 2012; *Mattern et al.*, 2012), and used to assimilate daily satellite observations of chlorophyll derived from ocean colour (see *Mannino et al.*, 2008).

The model's state space has more than 2 million dimensions and I evade problems posed by the high dimensionality by allowing errors in a few select biological parameters only. By ignoring model errors arising from other sources, such as physical model inputs, the procedure is effectively operating in a much lower dimensional error subspace. While only two biological parameters are selected to contain errors in this application, the error subspace approach can be easily extended to more parameters and other stochastic inputs. My goal in this chapter is to explore this particle filter configuration and identify effective approaches that allow me to obtain reliable and improved model chlorophyll and biological parameter estimates through data assimilation.

The approach for particle filter-based data assimilation I take in this chapter is motivated by previous research. In Chapter 3, the same ocean model and the same set of observations is used to perform data assimilation using *polynomial chaos*, a statistical emulator technique (introduced by *Wiener*, 1938). I obtained both chlorophyll state estimates and biological parameters estimates from the emulator. The model and observations from Chapter 3 are used as a suitable test bed for state and parameter estimation.

Results from the previous chapter also influenced the particle filter implementation. In Chapter 3, I found that the optimal values of the biological parameters exhibit a strong seasonal cycle and that these time-dependent parameters explain the data considerably better that the optimal but fixed parameters. This motivated my implementation of state-augmentation (Section 4.3.2) which permits time-dependent parameter estimation within the particle filter framework. In Chapter 3, I also found that the optimal parameter values vary strongly due to the presence of outliers and errors in the observations. These findings gave rise to my implementation of the asynchronous data assimilation (ADA; Section 4.3.4) and the implementation of the SIR weighting procedure (Section 4.3.1), both of which help to increase the robustness of the particle filter, allowing it to be effectively implemented for a realistic 3-dimensional biological ocean model using real satellite observations.

4.2 Methods

4.2.1 Particle Filtering Overview

Before presenting my particle filter implementation, a short overview of the basic particle filtering procedure is provided here. More detailed descriptions of particle filters and some of their extensions can be found in *Ristic et al.* (2004); *Dowd* (2007); *van Leeuwen* (2009).

For particle filters and ensemble-based data assimilation techniques in general, models are considered to be stochastic, i.e. the model state is represented by a multi-variate probability distribution. This probability distribution is approximated with an ensemble of particles. Each particle, or ensemble member, is a particular model state and represents a sample from the probability distribution of the model state. Starting with an initial ensemble of particles, sequential data assimilation techniques perform a sequence of assimilation steps to propagate the ensemble forward in time and update it with information from observations as they become available. Each assimilation step consists of two distinct sub-steps: the forecast step, and the observation update step.

The forecast step simulates the time evolution of the ensemble. Each particle is moved forward in time with the numerical model. For this purpose, the model state associated with each particle is used as an initial condition for the model and a model simulation is started for each particle. The model simulations are run until the next point in time with available observations and a new ensemble is formed from the most recent model states. In the subsequent observation update step, the new ensemble is updated with information from the newly available observations. This update typically accounts for observation errors as well as model uncertainty which is derived from the ensemble. While the forecast step is conceptually the same in all sequential techniques, including the EnKF, the filters differ in their implementation of the observation update step.

The SIR observation update step consists of a weighting of each ensemble member dependent on the current observations and a subsequent weight-based resampling. The weights assigned to each particle represent their distance from the current observations. In particle filter theory, the weight of the i^{th} ensemble member x_t^i given the current observation y_t is equal to the likelihood $p(y_t|x_t^{(i)})$ (the probability of the observations given the ensemble member). In practice, this likelihood is often unknown (and in case of high-dimensional model states, typically infeasible to compute) and thus the weights need to be approximated.

After the weights have been determined, a weight-based resampling of the ensemble is performed. An ensemble member with high fidelity to the observations, and therefore a large weight, has a higher probability of being drawn during the resampling than a lower-weighted particle. The sampling is performed *with replacement* so that particles can be drawn more than once. This means that at the end of the resampling the new ensemble will typically include multiple replicates of high-weighted particles, while some low-weighted particles do not get resampled (in other words, it is a weighted bootstrap). Through this procedure, the current observations have now been assimilated into the updated ensemble, and the ensemble-approximated model distribution moved towards the observations. The model-generated states that enter the observation update, leave it intact and remain true to the model dynamics, as the resampling introduces no potentially undesirable shifts in the ensemble members.

4.2.2 Model and Chlorophyll Observations

In this chapter, the particle filter is applied to the same 3-dimensional physical-biological ocean model introduced in Section 3.2.1. To enable direct comparisons between the particle filter and the emulator approach introduced in the previous chapter, the same set of satellite chlorophyll observations (see section 3.2.2) is used for particle filter assimilation as well.

Assimilation of daily images tests the particle filter's ability to deal with outliers and missing pixels. Large portions of the images can be missing due to clouds and other effects, so that some images contains only localized information about parts of the domain. In addition, significant noise is present in the satellite data set, especially close to the coast. As noted in Chapter 3, high noise levels and the abundance of dissolved organic material impede data assimilation and parameter estimation in the estuaries. For this reason, the estuaries are excluded from the assimilation experiments presented here.

4.2.3 Biological Parameters of Interest

An important part of the particle filter implementation, the ensemble generation, relies on treating two selected parameters of the biological model as stochastic. In this section I will briefly describe the role of the two parameters in the model and motivate why I selected these parameters in particular.

The first parameter of interest, θ_1, is the phytoplankton *maximum chlorophyll-to-carbon ratio*, it regulates the mechanism by which phytoplankton can adapt their chlorophyll content. The second parameter θ_2 is a *zooplankton grazing* parameter which controls the rate at which zooplankton consume phytoplankton. While θ_1 affects the chlorophyll variable in the model directly, θ_2 has a more indirect effect on chlorophyll, through altering the grazing pressure on phytoplankton and hence their abundance. My motivation for selecting these two parameters is twofold: firstly, the model exhibits a strong sensitivity to relative changes in both parameters, especially θ_1. Consequently, even small parameter adjustments can have a large effect on the chlorophyll output of the model. Secondly, both parameters are physiological plankton parameters known to vary greatly among species and can thus be considered stochastic with a high uncertainty. Varying these parameters presents a suitable way to produce variable chlorophyll output in the model. While it would be easy to incorporate more parameters into the particle filter procedure, I limit the analysis to these two parameters because their effects were examined and their optimal values were identified in Chapter 3. This allows me to compare and contrast the particle filter-based parameter estimates to previous results.

4.3 Particle Filter Implementation

In this section, I describe the aspects of the particle filter implementation that are important to this application and deserve special attention. While the SIR resampling is straightforward to implement, the computation of the weights can be implemented in various ways and needs careful consideration.

In this application, each particle in the ensemble is a realization of the biological state of the system described in Section 4.2.2, consisting of the concentrations of the 7 biological state variables in each of the 82×130 (horizontal)$\times 36$ (vertical) $= 383760$ grid cells of the model. As the state of the physical ocean model is unaffected by the biological parameters, it remains deterministic and does not need to be considered by the SIR procedure or even included in the state vector. With only the biological variables included, a state vector is $7 \times 383760 = 2686320$ dimensional. For some of my particle filter experiments, I additionally append two model parameters to the state, this is described in detail in Section 4.3.2.

4.3.1 SIR Weighting

In the resampling step of the SIR procedure, a weight is assigned to each ensemble member proportional to its likelihood (see Section 4.2.1). In practice, the likelihood is typically unknown and can only be approximated. In my particle filter implementation, I base the computation of the associated weights on the assumption that the likelihood is inversely related to a suitable distance measure for model state and observations.

In this application, each observation is a satellite image, typically more than 1100 surface chlorophyll concentration values arranged in a matrix which also contains missing values (on average, the proportion of missing values is 75%). The corresponding surface chlorophyll field in the model can easily be transformed into a similar image at the same resolution, reducing the model-data comparison to the problem of comparing two images

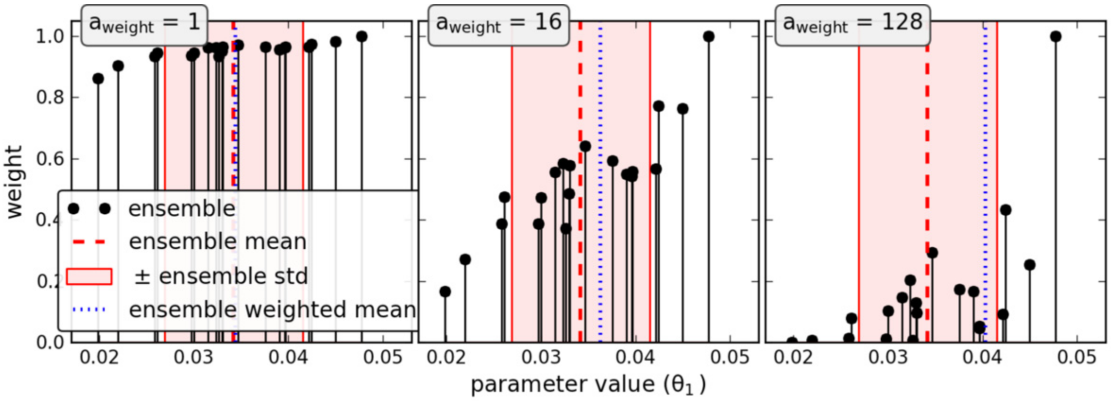


Figure 4.1: A typical ensemble of values for θ_1 illustrating the effect of the weighting parameter a_{weight}. Black dots show the weight of ensemble members during resampling over the corresponding parameter value for a weighting parameter a_{weight} of 1 (a), 16 (b) and 128 (c). Weights are obtained according to the procedure described in Section 4.3.1. Also shown are the ensemble mean and standard deviation of θ_1 (dashed and solid red lines) and the weighted ensemble mean (dotted blue line). The distance between weighted and unweighted ensemble mean acts as an indicator for the ensemble shift that can be expected from the SIR resampling. For $a_{\text{weight}} = 1$ the ensemble is not likely to shift significantly after resampling, due to the small differences in weights. By increasing a_{weight}, differences in weight become more pronounced, increasing the expected ensemble shift. For $a_{\text{weight}} \leq 128$, one or just a few ensemble members carry high weights, increasing the chance of ensemble collapse.

which typically contain missing values. For the purpose of image comparison, I employ the adapted grey block (AGB) distance measure introduced by *Mattern et al.* (2010b) which I also used in Chapter 3. To compute the AGB distance value, two images are compared at different resolution levels by dividing them into successively smaller blocks and computing the mean value of the pixels within each block. The root mean square error (RMSE) is then computed at each resolution level, from the highest resolution, where each block consists of only one pixel, to the lowest, where a single block encompasses the entire image. For the purpose of comparing satellite images, this multi-resolution approach has proven to be advantageous over standard approaches such as the regular RMSE (*Mattern et al.*, 2010b) because it is less sensitive to noise and adapted to work with missing values in the images. However, my SIR weighting implementation is flexible and any suitable model-data distance measure could be used.

To transform the AGB distance values into weights, I use the following procedure:

(1) Compute $d_i = \mathrm{d}_{\text{AGB}}(x_t^{(i)}, y_t)$, the AGB distance between the current satellite observation and ensemble member i for each $i = 1, \ldots, n_{\text{ens}}$.

(2) Calculate raw weights as the inverse of the AGB values: $\hat{w}^{(i)} = \frac{1}{d_i}$.

(3) Normalize the raw weights and transform them according to: $\tilde{w}^{(i)} = \left(\frac{\hat{w}^{(i)}}{\max_j \hat{w}^{(j)}}\right)^{a_{\text{weight}}}$, where a_{weight} is an adjustable parameter that spreads (or contracts) the distribution of weights (discussed in more detail below).

(4) Determine final weights as $w_t^{(i)} = \frac{\tilde{w}_t^{(i)}}{\sum_{j=1}^{n_{\text{ens}}} \tilde{w}_t^{(j)}}$, so that their sum is one.

The parameter a_{weight} in step (3) adjusts the weights before they enter the resampling process. It does not change their relative ranking but affects their variance. The parameter a_{weight} brings the weights closer together (for $a_{\text{weight}} < 1$) or spreads them apart (for $a_{\text{weight}} > 1$). In practice, such an adjustment is needed if the values of the weights are too similar or differ by orders of magnitude. The former is the case in this application: typical values of d_i (step 1) are strongly dependent on the satellite observation but do not vary much across the ensemble of model simulations. Consequently, the normalized raw weights obtained in step (2) are tightly clustered (more than 70% of the raw weights are between 0.9 and 1.0 during a typical assimilation run). If no adjustment is made to the weights before resampling takes place (beside the normalization in step (4)), the ensemble members are resampled at almost equal probability. As a result, the effect of the resampling on the ensemble is very small. This is illustrated by the example in Figure 4.1. For $a_{\text{weight}} = 1$, which corresponds to no weight adjustment, the SIR resampling will have almost no effect even in cases where parameter and weights are strongly correlated, i.e. an increase in θ_1 is accompanied by an increase in weights (Figure 4.1(a)). For higher values of a_{weight}, ensemble members with higher weights become more prominent (Figure 4.1(b)), up to a point where one ensemble member dominates the ensemble (Figure 4.1(c)). In the latter case the ensemble is likely to collapse to just a few unique particles. The adjustment of a_{weight} has a similar effect as changing the observation error distribution, which is done, for example, to decrease the effects of outliers (*van Leeuwen*, 2003). In this application, both the true likelihood and the value of a_{weight} that would best approximate the likelihood are unknown. Furthermore, I am limited to a small number of particles, so that it is essential to select a_{weight} to extract a maximum amount of information from the observations while avoiding ensemble collapse – even though this step introduces subjectivity into the weighting procedure. I set $a_{\text{weight}} = 16$ for this application, based on a series of experiments in which I simulated thousands of assimilation steps, measured the average raw ensemble weight after resampling (see step(2)) and estimated the probability of ensemble collapse.

The effects of this weighting implementation on the ensemble structure and its development from one assimilation step to the next are visualized in Figure 4.2 for a typical SIR simulation with 10 ensemble members. It gives a detailed view of the time history of the particles and the effect of the weights on the resampling of the ensemble. By tracing each particle back to its "parent" particle, the particle it was resampled from, I determine that 7.5 is the average number of assimilation steps it takes to trace a given ensemble back to its last common "ancestor" particle (see green line in Figure 4.2). This number indicates that the expected number of ensemble collapse events is relatively low and that particles with high likelihoods do not get too much weight. At the same time it is apparent from Figure 4.2 that the weighting is effective at steering the ensemble.

4.3.2 State-Augmentation

State-Augmentation is an extension that permits parameter estimation within the particle filter framework and was introduced by *Kitagawa* (1998). In this extension, the model state x_t is augmented by a vector of model parameters, $\theta_t \in \mathbb{R}^{n_\theta}$ to form

$$x_t^\star = \begin{pmatrix} x_t \\ \theta_t \end{pmatrix}.$$

The new, augmented state vector replaces the regular state vector in the particle filter procedure (Section 4.2.1) so that the ensemble of state vectors now carries information about parameter values, which are weighted and resampled along with all other information contained in the vector. The process of resampling parameter values yields a time-dependent parameter distribution which can adapt to improve the fit to the observations. Suitable parameter values produce good state estimates which are characterized by being similar to the observations. This, in turn, results in high likelihood values for state vectors with suitable parameter values, ensuring that they are resampled with a higher probability than less suitable ones. As a result, after a number of resampling steps, the ensemble should be populated by the parameter values that best fit the observations (*Dowd*, 2011).

In this application I augment the state vector by $n_\theta = 2$ parameters. As described in

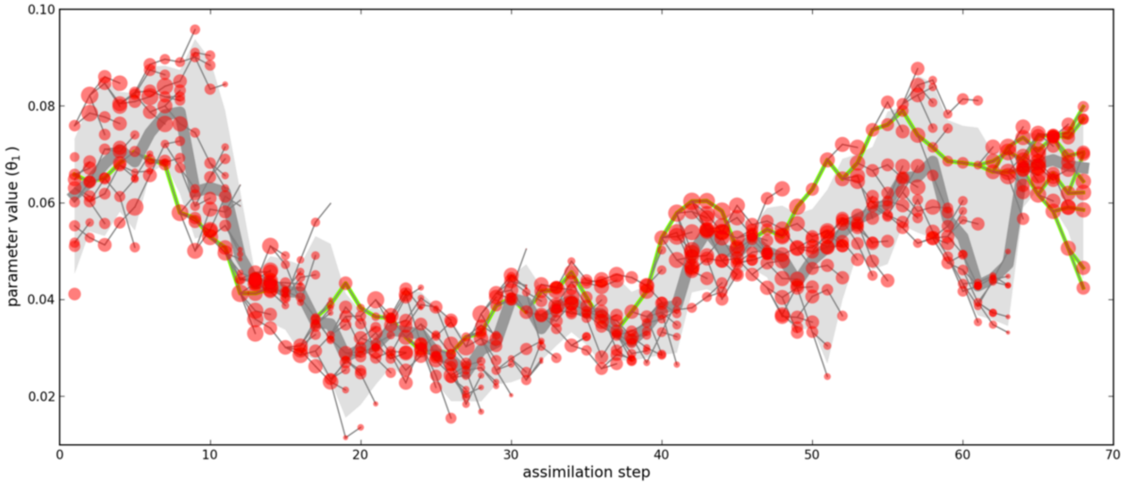


Figure 4.2: A detailed view of the development of θ_1 during a typical particle filter experiment with an ensemble size of $n_{\text{ens}} = 10$ over 68 assimilation steps. Each red dot marks one particle, the area of the dot is proportional to its weight during the resampling. The thin lines connect each particle to its "parent" particle from which it was resampled and its "children" which are resampled from it. The lines highlighted in green trace the last ensemble at assimilation step 68 back through their ancestors. The (vertical) parameter value offsets between a parent particle and its children are due to the parameter noise that is applied during resampling. The ensemble median and the region between the 0.1 and 0.9 ensemble quantiles are displayed in the background (dark grey line and light grey area, respectively). This, less detailed view is used in Figures 4.5 and 4.6. The data for this plot is taken from one of the particle filter experiments in Section 4.4.5, it uses the standard particle filter configuration.

name	description	optimal value	initial distribution	truncated distribution[a]	parameter perturbation sd[b]
θ_1	maximum Chl:C ratio in phytoplankton	0.047	$\mathrm{U}(0.013, 0.094)$	$\mathrm{U}(0.040, 0.094)$	0.004
θ_2	zooplankton grazing parameter	0.650	$\mathrm{U}(0.150, 1.050)$	$\mathrm{U}(0.450, 1.050)$	0.045

[a] Used in the state-augmentation experiment in Section 4.4.4.

[b] Applied to θ_1 and θ_2 after the particle filter resampling; equal to $\frac{1}{20}$ of the range of its initial distribution.

Table 4.1: Optimal parameter values and other properties for the particle filter experiments.

Section 4.2.3, these parameters are two biological parameters, the maximum carbon-to-chlorophyll ratio (θ_1) and the maximum zooplankton growth rate (θ_2). Initial values for the parameters are drawn from parameter specific uniform distributions, which are listed in Table 4.1 and based on parameter ranges used in Chapter 3. In my implementation of the particle filter procedure, the parameters in the state vector are treated the same way as all other entries in $x_t^\star$, with the exception that perturbations of the model state only affect the parameters.

4.3.3 Ensemble Generation and the Error Subspace

Perturbations to the state vectors are necessary in my particle filter setup to prevent the ensemble from degenerating. If no perturbations are introduced after resampling, two replicates of the same ensemble member would result in two identical model simulations, since both model state and parameter values are resampled together. Over the course of multiple assimilation steps, replicates of ensemble members with high weights could replicate further until the ensemble degenerates into a collection of one or just a few unique ensemble members. This is referred to as ensemble degeneracy or ensemble collapse and presents a well-known issue with particle filters (*Dowd*, 2006). Many schemes have been developed to address this problem (*van Leeuwen*, 2009). Here, I simply perturb the parameters that are part of the augmented state. This approach avoids modifications to the rest of the state vector, thus keeping the resampled model states true to the model. While the model contains other error sources, introduction of noise through parameters and other model inputs is an effective way to add uncertainty into the model results, and is explored in more detail in the following chapter. This approach has the additional benefit of avoiding shocks to the biological model by creating consistent biological states.

In my implementation, I vary the augmented parameters by adding normally distributed random noise to the parameter values after they have been resampled. The standard deviation of the noise is parameter-specific and listed in Table 4.1. In the rare cases where the noise causes a parameter value to become negative, the sign of that value is switched, ensuring positive parameter values. No measures are taken to prevent parameter values from exceeding any upper limit. From an implementation point of view, it is important that enough noise is added to each parameter to effectively counteract ensemble degeneracy, but not so much that state-augmentation becomes meaningless. That is, if the addition of noise causes the parameter values to deviate far from their original values, the positive effect of resampling the parameters will be lost.

The evolution of the ensemble and its error sources can be summarized in a state-space equation. With state-augmentation, the state-space equation of this application is

$$\begin{pmatrix} x_{t+1} \\ \theta_{t+1} \end{pmatrix} = \begin{pmatrix} F_{\text{model}}(x_t, \theta_t) \\ \theta_t \end{pmatrix} + \begin{pmatrix} 0 \\ \epsilon_\theta \end{pmatrix},$$

where F_{model} represents the model which propagates the biological state x_t from one assimilation time step to the next, using the parameters in θ_t. The parameter vector θ_t is only modified by the additive error term ϵ_θ (as described above), which allows it to adapt to the observations through the filter-based estimation procedure. Unlike the parameters, no error is introduced to the biological state directly.

In this application, I know that time varying values of θ_1 and θ_2 can describe the data significantly better than the optimal but fixed parameter set (see Chapter 3). To allow the particle filter to capture the time dependence of suitable parameter values, I choose a relatively high standard deviation for the parameter noise ϵ_θ, permitting the parameter values to easily readjust in the course of assimilation (Figure 4.2).

The variation of θ_1 and θ_2 gives rise of the idea of an error subspace: all of the error introduced into the ensemble originates in perturbations of these two parameters. Therefore, the particle filter ensemble needs to approximate the 2-dimensional parameter distribution, rather than the much higher dimensional distribution of the state space. As a consequence, I may expect good representation of the distribution even for a relatively small ensemble.

One shortcoming of this approach is that it is based on the assumption that the model error can be explained by a few parameters (here: θ_1 and θ_2) alone, although there are numerous other error sources (e.g. other parameters, boundary and initial conditions, or the model's spatial and temporal discretization) – a point that is discussed in more detail in Section 4.5. However, good results can be expected from the error subspace approach if the parameters that are varied are major sources of model error that can induce enough variability into the ensemble to avoid its degeneration. From Chapter 3, I know that values of θ_1 and θ_2 vary greatly and also that they have a strong impact on the observed variable chlorophyll. Thus, the two parameters fulfil the requirement of being major error sources with regard to model chlorophyll.

4.3.4 Asynchronous Data Assimilation

Particle filters that assimilate observations sequentially, and one-by-one, are typically very sensitive to outlying observations which can induce strong, undesired shifts of the ensemble. To increase the robustness of the particle filter in the presence of outliers, data can be assimilated asynchronously, i.e. not individually as each observation becomes available, but as a set of observations at a later point in time. Many implementations of asynchronous data assimilation have been developed for sequential techniques, e.g. for particle filters (*Godsill et al.*, 2004; *van Leeuwen*, 2009; *Dowd*, 2011) and the EnKF (*Evensen and van Leeuwen*, 2000; *Sakov et al.*, 2010) and are often referred to as *fixed lag smoothers* as they are meant to smooth the evolution of the ensemble and remove undesired shifts. The general idea is to run the model forward, collect several observations and assimilate them into the model together.

Here, I implemented a simple asynchronous data assimilation (ADA) scheme for the particle filter mainly because the noise level in the satellite observations is high and they contain a large number of missing values. The latter may cause a single image to contain only localized information about part of the domain. Both factors contribute strongly to the rapid changes of optimal parameter values (see Figure 3.5).

The ADA procedure is as follows:

(1) Initialize the counter variable $n_{\text{counter}} = 0$ and start the regular SIR particle filter procedure.

(2) Proceed to the next forecast step and increase n_{counter} by 1.

(3) If $n_{\text{counter}} \leq n_{\text{ADA}}$, compute the weights for the current observation but skip SIR resampling, and continue to step (2).

(4) Otherwise, compute the weights for the current observation and collect the n_{ADA} previous weights, average them for each ensemble member and use the average weight to perform SIR resampling. Continue to step (1).

Note that the model does not need to be stopped and restarted when no resampling occurs. It is sufficient to store model output at the time steps corresponding to the available observations, so that the weights can be computed when a full assimilation step is performed. The above procedure represents a simple way to combine $n_{\text{ADA}}+1$ observations

	name	section	n_{ens}	S-A[a]	ADA	n_{ADA}	trunc. dist.[b]	n_{rep}[c]
	reference simulation	4.4.1	*the reference simulation does not use SIR*					
1	baseline	4.4.2	20	✓	✓	4		9
2.1	no ADA	4.4.3	20	✓		–		3
2.2	strong ADA	4.4.3	20	✓	✓	9		5
3.1	no state-augmentation	4.4.4	20		✓	4		1
3.2	truncated baseline	4.4.4	20	✓	✓	4	✓	1
3.3	truncated, no S-A[a]	4.4.4	20		✓	4	✓	1
4.1	$n_{\text{ens}} = 10$	4.4.5	10	✓	✓	4		3
4.2	$n_{\text{ens}} = 5$	4.4.5	5	✓	✓	4		4

[a] state-augmentation
[b] using the truncated distribution
[c] number of replicates

Table 4.2: Configuration of the particle filter experiments.

into one assimilation step. Outlying observations still enter the SIR resampling but their effect on the ensemble is greatly reduced by the averaging process. The effect of this ADA scheme are assessed in Section 4.4.3.

4.4 Experiments and Results

4.4.1 Particle Filter and Model Setup

I use the following basic setup for all experiments. Unless noted otherwise, my experiments are particle filter simulations for the year 2006. I start the model simulations on January 1st, after an adjustment phase of 10 days the 350 daily satellite observations are assimilated into the model. The model simulations are stopped on December 31st. The initial conditions for the physical and biological variables are taken from a spun up model simulation started January 1st 2005. Initial parameters are taken from the distributions listed in Table 4.1.

To assess the particle filter results quantitatively, I compare them to a model simulation with optimized, fixed parameters. This reference simulation uses the same model with an identical setup and optimal parameter values obtained from the emulator approach in Chapter 3. The time-dependent optimal values for θ_1 and θ_2 are based on the same set of observations that I use for the particle filter assimilation (including the removal of the estuaries data, see Section 4.2.2). This means that the reference simulation produces the

	average absolute residuals			
name	*shelf* region[a]	*open ocean* region[a]	April, *open ocean*	December, *open ocean*
reference simulation	0.6276	0.2413	0.5795	0.4000
particle filter baseline	0.6124 ± 0.0554	0.2532 ± 0.0148	0.5572 ± 0.1163	0.3562 ± 0.0108
$\hookrightarrow$ worst[b]	0.7185	0.2726	0.7219	0.3833
$\hookrightarrow$ best[b]	0.5537	0.2325	0.3722	0.3486
no ADA	0.6706 ± 0.1008	0.2391 ± 0.0206	0.3889 ± 0.0156	0.3613 ± 0.0063
strong ADA	0.6763 ± 0.0629	0.2986 ± 0.0209	0.9185 ± 0.2113	0.3727 ± 0.0231
no state-augmentation	0.6610	0.2602	0.7248	0.3813
$n_{\text{ens}} = 10$	0.6236 ± 0.0204	0.2523 ± 0.0140	0.5280 ± 0.0813	0.3761 ± 0.0076
$n_{\text{ens}} = 5$	0.6424 ± 0.0042	0.2798 ± 0.0286	0.8419 ± 0.2378	0.3772 ± 0.0265
truncated baseline	0.5830	0.2431	0.4836	0.3566
truncated, no S-A[c]	0.8032	0.3442	1.0120	0.3451

[a] See Figure 4.4 for a map of the regions.
[b] The residuals of the worst and best results of the baseline experiment in each column are taken from different simulations.
[c] state-augmentation

Table 4.3: Absolute residuals (mean ± standard deviation) for the particle filter experiments.

best chlorophyll output I can expect from a regular simulation of the model in this scenario. If the particle filter can achieve similar results to the optimized reference simulation, I consider it a success (see Table 4.2 for an overview of all experiments).

In the following, I introduce the particle filter baseline experiment (Section 4.4.2), a simulation that uses the particle filter with asynchronous assimilation and state-augmentation. In the experiments thereafter, I assess the individual effects of ADA (Section 4.4.3) and state-augmentation (Section 4.4.4). Finally, I explore the effect of the ensemble size on the baseline experiment (Section 4.4.5).

Because the particle filter results are based on parameters that are drawn from a uniform distribution initially and assigned random noise terms, I perform several replicate simulations to assess the stability of the particle filter experiments and the reliability of its results. The replicates simulations are exact copies of the particle filter experiments, except for the random terms. For experiments with replicates runs, the results listed in Table 4.3 show the mean result and its standard deviation. In the table, I also included the best and the worst results of the baseline experiment in order to provide a full account of the results encountered in the replicate simulations.

4.4.2 Experiment 1: Baseline Experiment

The first particle filter experiment uses the standard configuration and acts as a baseline that other configurations are compared against. It uses ADA (with $n_{\text{ADA}} = 4$), state-augmentation, and an ensemble size of $n_{\text{ens}} = 20$.

In order to compare the performance of the baseline experiment to the reference simulation, I focus on the average surface chlorophyll content in the *shelf* and *open ocean* regions of the model domain (Figure 4.4). The reference simulation recreates the chlorophyll dynamics in the observations reasonably well (Figure 4.4(a)). The best fit is achieved in the months following the chlorophyll spring bloom (May, June and July) when chlorophyll levels stay relatively low and do not fluctuate much. The biggest discrepancy appears during the bloom in April in the *open ocean* region where the model overestimates the surface chlorophyll content, and late in the year in December when chlorophyll is underestimated. Because there is most room for improvement in these months, I include the mean absolute chlorophyll residual for April and December in Table 4.3, along with the mean absolute residuals for the entire year.

In comparison to the reference simulation, the baseline particle filter configuration

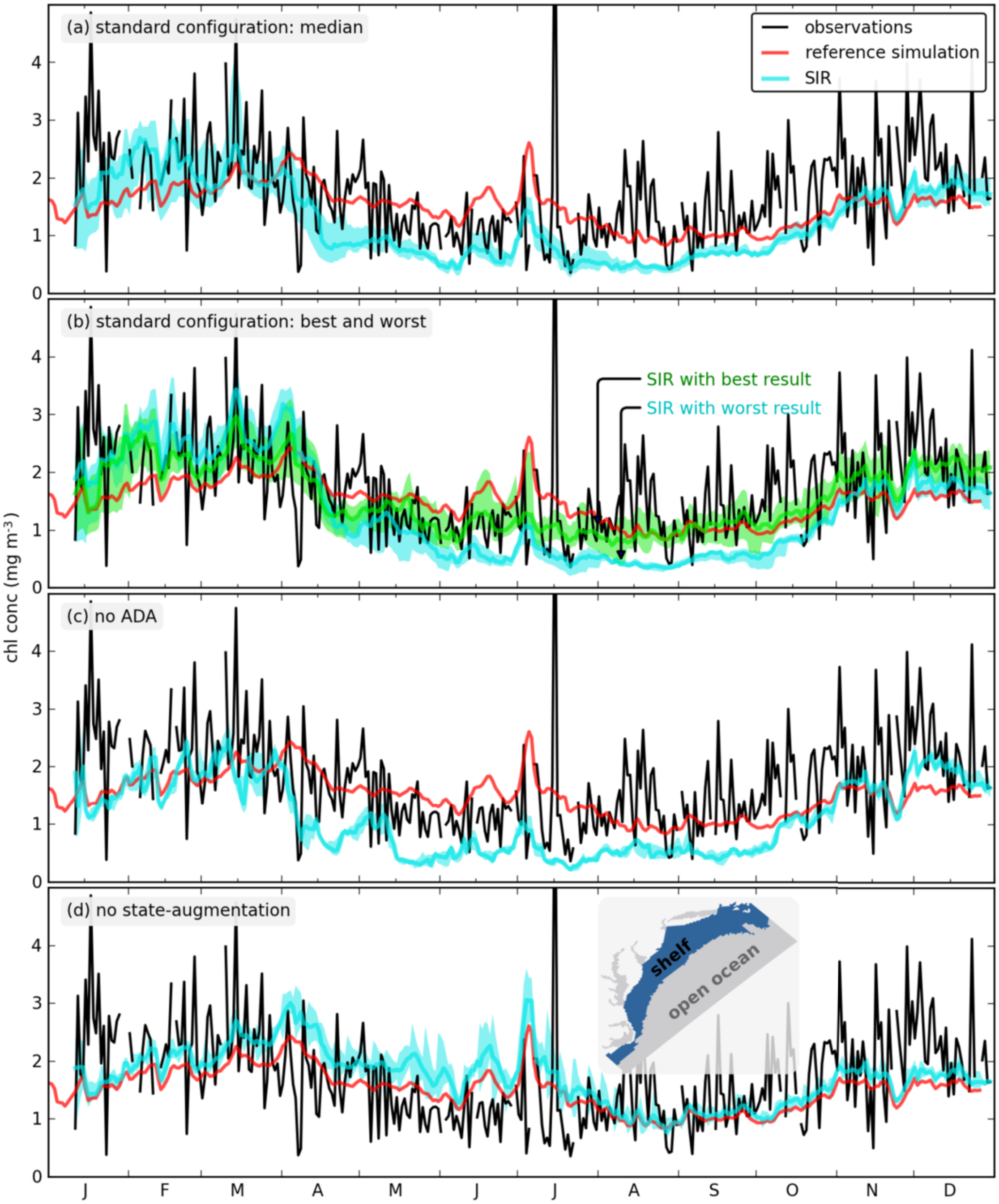


Figure 4.3: The surface chlorophyll development in the *shelf* region (see inset in (d)) for selected particle filter experiments in comparison to the observations and the reference simulation with optimized parameters. For the particle filter, both ensemble median and the region between the 0.1 and the 0.9 quantile are shown. The panels correspond to different particle filter experiments. The particle filter simulation that achieved the median fit to the observations among the replicates of the baseline experiment (Section 4.4.2) is shown in (a), those with the worst and best fit are displayed in (b). An experiment without ADA (Section 4.4.3) is depicted in (c), an experiment without state-augmentation (Section 4.4.4) in (d).

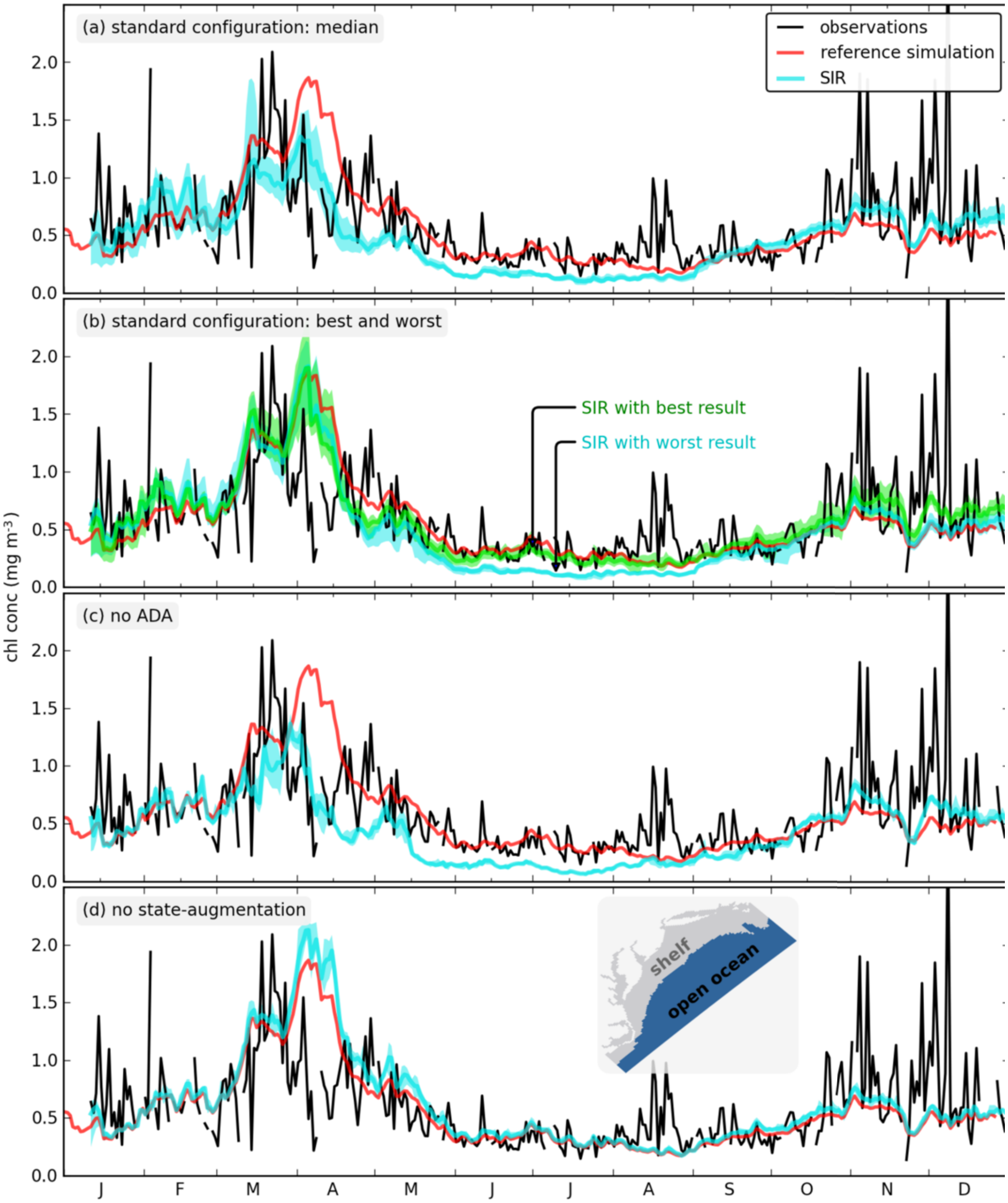


Figure 4.4: The surface chlorophyll development in the *open ocean* region (see inset in (d)) for the particle filter experiments in Figure 4.3. For the particle filter, both ensemble median and the region between the 0.1 and the 0.9 quantile are shown. The panels correspond to different particle filter experiments. The particle filter simulation that achieved the median fit to the observations among the replicates of the baseline experiment (Section 4.4.2) is shown in (a), those with the worst and best fit are displayed in (b). An experiment without ADA (Section 4.4.3) is depicted in (c), an experiment without state-augmentation (Section 4.4.4) in (d).

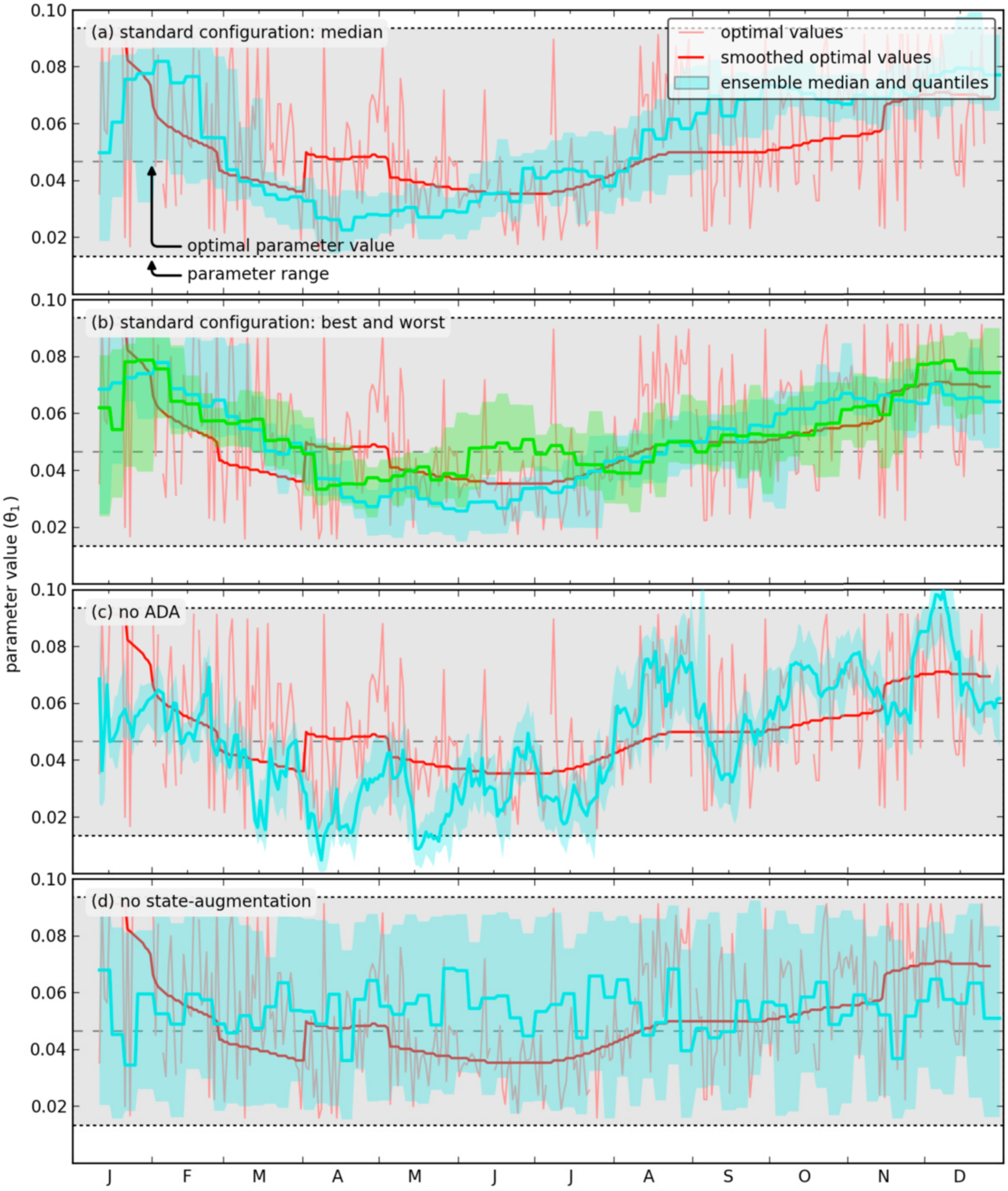


Figure 4.5: The development of θ_1 for the same experiments depicted in Figure 4.4. The faint red line and the thick red line correspond to the optimal parameter values obtained in Chapter 3. For the particle filter, both ensemble median and the region between the 0.1 and the 0.9 quantile are shown.

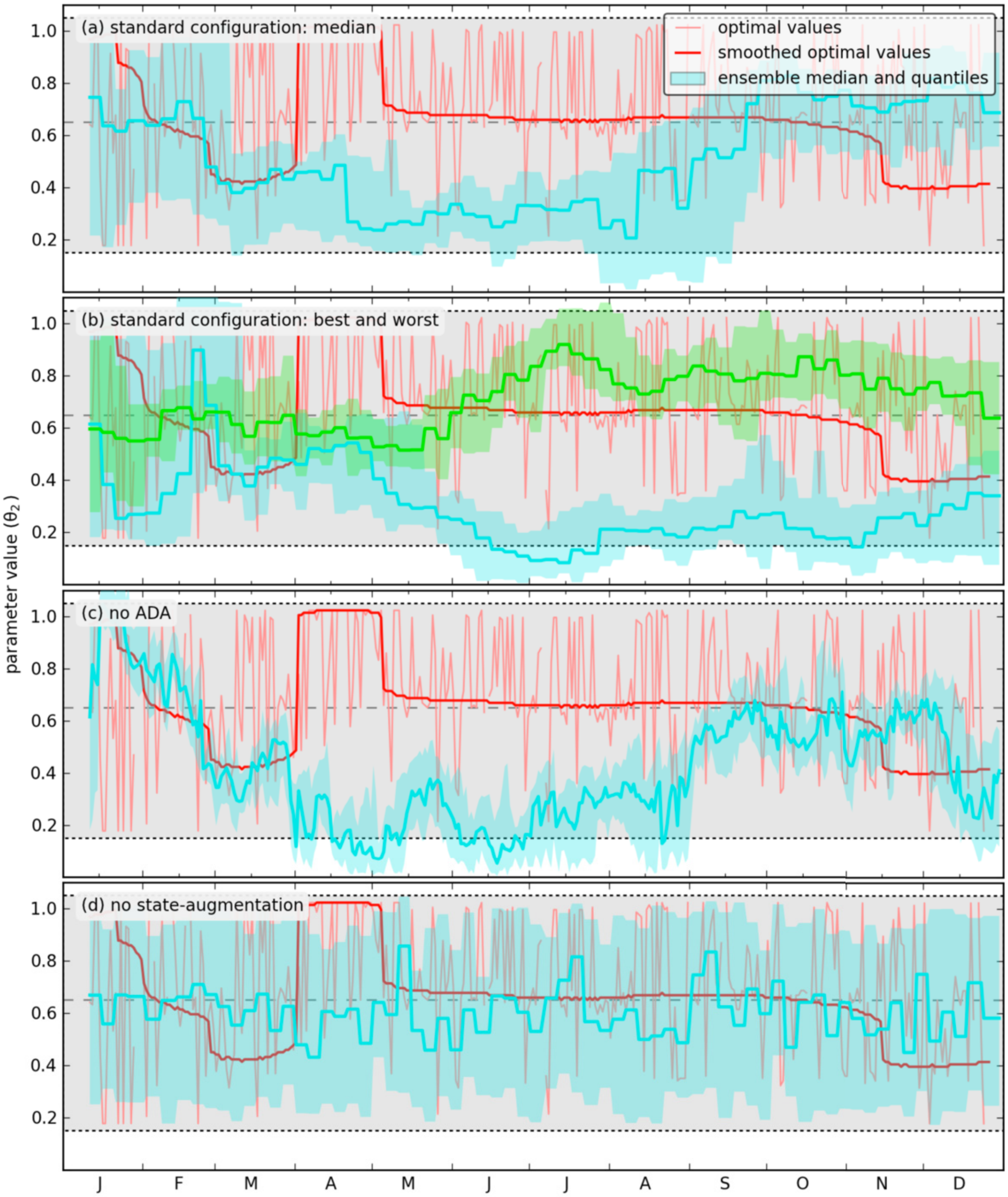


Figure 4.6: The development of θ_2 for the same experiments depicted in Figure 4.4, corresponding to the development of θ_1 in Figure 4.5.

performs, on average, slightly better in the *shelf* region and slightly worse in the *open ocean* region (Table 4.3). For a better overview, I have included the simulations that achieved the worst, the best and the median results among replicate simulations (based on absolute residuals in the *open ocean* region) in Figure 4.4 and 4.5. There are some obvious differences between individual simulations that use the same configuration: the median simulation (Figure 4.4(a)), for example, shows an improved fit to the observations in April, then a underestimation of chlorophyll throughout August, and finally improvement again towards the end of the year. In contrast, the best simulation does a worse job at predicting the decline of the chlorophyll bloom in early April but performs better for the rest of the simulation (Figure 4.4(b)).

Despite obvious differences among individual simulations, even the worst particle filter results remain close to those of the reference simulation (see Figure 4.4(b)). The highest variation can be found in the April chlorophyll residuals (Table 4.3), where most but not all particle filter simulations can improve upon the reference simulation and where the best results show a very strong improvement. In contrast, all particle filter simulations can improve upon the reference simulation for the December residuals and the standard deviation among simulations is considerably smaller. In the following section, I examine if the relatively low variation of the particle filter residuals throughout most of the year is due to the stabilizing effects of the ADA.

The parameter development for the particle filter with standard configuration (Figure 4.5(a) and (b) for θ_1 and Figure 4.6(a) and (b) for θ_2) reveal that the values of the augmented parameters do not converge to a fixed value but roughly retrace the smoothed optimal time-dependent values for θ_1. This means that the assimilation can capture the time-dependence of θ_1, the parameter with the strongest effect on model chlorophyll. For θ_2, which has a smaller impact on surface chlorophyll (Figure 3.3), optimal parameters are retraced less closely, but the 2-dimensional parameter distribution remains centered in a region that produces adequate results. The fact that the particle filter simulations can identify the time-dependence and follow the optimal parameter values likely contributes to improved chlorophyll estimates. To what extent the chlorophyll estimates benefit from state-augmentation is investigated in Section 4.4.4.

4.4.3 Experiment 2: Effect of ADA

In my second experiment, I perform one simulation where ADA is deactivated, so that each observation is assimilated individually and one simulation where I increase the effect of ADA by increasing the smoothing window and setting n_{ADA} to 9.

The surface chlorophyll development for the experiment without ADA (Figure 4.4(c)) shows some interesting features in comparison to the baseline experiment. The average, year-long absolute residuals and their variability among the experiments have notably increased without ADA (Table 4.3). The particle filter has become less robust and the general performance over the time span of the entire experiment has dropped. Yet, the April and December residuals exhibit an improvement compared to the baseline experiment and the associated variability during these months is lower too. One possible explanation for these results can be found in the evolution of the parameter values (Figure 4.5(c)). In instances where there is a large deviation between the model and the chlorophyll observations, the daily assimilation can quickly adjust model state and parameter values. This can be advantageous in the short term (April and December) where it can lead to a more consistent improvement compared to the standard experiment. Yet strong shifts also occur if the particle filter estimates are close to the observations. These shifts have a negative impact and frequently force the parameter estimates away from the optimal parameter values. For example, a strong shift in θ_1 in May (see Figure 4.5(c)) causes a long-lasting underestimation of chlorophyll (Figure 4.4(c)). Overall, the negative shifts outweigh the positive ones, evidenced by the year-long results in Table 4.3.

One additional problem that affects the simulations without ADA is caused by the daily assimilation. The simulations without ADA exhibit a low ensemble spread (compare Figure 4.4(c) to (a) or (b)) which hinders a quick recovery from shifts in the ensemble. The low spread is likely caused by the faster succession of assimilation steps (daily compared to every 5 days in the baseline experiment) which contract the ensemble more frequently. Furthermore, the use of ADA decreases the number of independent observations that are assimilated. More independent observations, in the simulations without ADA, have a negative effect on particle filter performance, as they can increase the probability of ensemble collapse (*Snyder et al.*, 2008; *van Leeuwen*, 2009).

After deactivating ADA in one simulation, I perform a further simulation to explore the effect of increasing n_{ADA}, the number of observations that are averaged in the ADA procedure. For $n_{\text{ADA}} = 9$, the result is a slight degradation of the mean performance compared to the baseline experiment (Table 4.3). Here it appears that the wider gap between assimilation steps prohibits a quick adjustment of the ensemble causing the assimilation to become sluggish. Given these results, I expect no increase in performance for $n_{\text{ADA}} > 9$.

In summary, the ADA has a stabilizing effect on the particle filter simulations. Without ADA, resampled parameter values appear less smooth in time and the particle filter state estimates become less reliable due to outliers in the data. By averaging the weights over multiple time steps with the ADA procedure, I can reduce the problem posed by outliers in the observations while still permitting parameter values to change in time.

4.4.4 Experiment 3: Effect of State-Augmentation

In this experiment I examine the effects of state-augmentation on the particle filter assimilation results. State-augmentation serves as a way to estimate parameter values with particle filters and provides the ensemble with a parameter memory: suitable parameter values are "remembered" by carrying them over from one assimilation step to the next. The base run with state-augmentation exhibits a clear pattern in the ensemble of parameter values, retracing the optimal time-dependent parameter values (Figure 4.5(a)). This indicates that a memory of suitable parameter values is advantageous and may be required to obtain agreement of the chlorophyll estimates.

To assess the influence of state-augmentation, I perform a particle filter simulation with the same configuration as the baseline experiment, but deactivate state-augmentation. Parameter values are now no longer resampled but instead redrawn from a static distribution at the end of every assimilation step, rendering the parameter distribution essentially memoryless (illustrated in Figure 4.5(d)). In an initial experiment, I use the same uniform distributions that the parameters are drawn from initially. To assess the influence of the parameter distribution I perform an additional experiment replacing the original distribution with a truncated one (see Table 4.1).

Without state-augmentation, the ensemble median of the particle filter simulation remains close to the reference simulation (compare Figure 4.4(d)) and consequently the results of the two simulations are very similar (Table 4.3). The resampling of states in combinations with random assignment of parameter values seem to have little effect on the

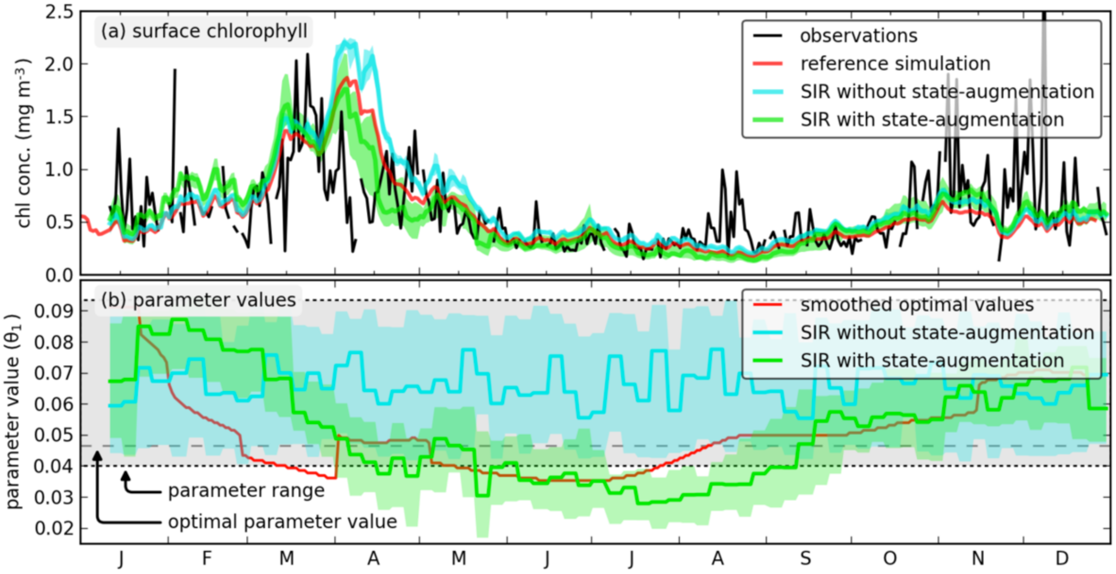


Figure 4.7: The development of the surface chlorophyll concentration (a) and θ_1 (b) for the particle filter with truncated parameter distributions (compare Section 4.4.4) with and without state-augmentation. In both panels, the ensemble median and the region between the 0.1 and the 0.9 quantile are shown for both particle filter experiments. Panel (a) also contains the observations and the reference simulation, (b) shows the smoothed optimal values (obtained from Chapter 3).

evolution of the ensemble. In fact, in the case of the average surface chlorophyll concentration, the ensemble median remains above the output of the reference simulation throughout the entire simulation. This bias is present even when it implies a severe overestimation of the chlorophyll concentration (see for example the April result in Figure 4.4(d)). In this particle filter configuration, the resampling cannot effectively correct chlorophyll estimates which instead appear to be dependent on the static parameter distribution chosen for the experiment.

To further assess the influence of the parameter distributions, I perform an additional experiment with modified distributions. Again, I use uniform distributions but truncate their range by one third on the upper side (see Table 4.1). As a result, the optimal parameter values are no longer close to the mean of their respective distributions. The use of these new distributions adds a bias to the chlorophyll estimates and causes a clear decline in the fit to the observations for the particle filter without state-augmentation (Fig 4.7). With state-augmentation turned on, however, the effects of the ill-chosen initial parameter distribution are barely noticeable, as the resampling quickly determines the time evolution of parameter values (Fig 4.7(b)). This experiment clearly confirms my previous observation that the particle filter configuration without state-augmentation is strongly dependent on parameter distribution, whereas the resampling of states has a negligible corrective effect: a closer investigation reveals that the particle filter implementation without state-augmentation produces on average considerably lower weights during resampling when compared to particle filter with state-augmentation. Due to the random assignment of parameter values, none of the particles in the small ensemble experiences a lasting benefit from suitable parameter values. This negative effect cannot be effectively counteracted by the SIR resampling.

It is apparent that the parameter memory provided by state-augmentation is an important asset in this application. It allows the particle filter to resample parameter values and focus them in suitable regions in parameter space which are allowed to vary in time. In addition, good results can be obtained without prior knowledge of suitable parameters values, as state-augmentation provides the particle filter with a means to adjust parameters quickly and break out of ill-posed initial distributions.

4.4.5 Experiment 4: Ensemble Size

The ensemble represents a sample from the probability distribution of the biological state variables. It is expected that a larger ensemble and therefore a higher sample size should lead to a better representation and more accurate results. However, computational cost of an particle filter simulation grows linearly with the ensemble size and computational constraints force me to keep the ensemble size small. In my final experiments, I examine the influence of the ensemble size on the particle filter implementation.

Since the standard particle filter configuration yielded good results with $n_{\text{ens}} = 20$ ensemble members, my first experiments involve a decrease in the ensemble size. For $n_{\text{ens}} = 10$ I can detect no significant difference in my results in Table 4.3 in comparison to the standard configuration. A further reduction of the ensemble size to 5, however, causes a decline in performance. The relatively high standard deviation of the SIR results with 20 ensemble members and the differences between the best and worst results in Table 4.3 indicate that ensemble sizes larger than 20 may lead to more stable results. However, due to the slow convergence rate of Monte Carlo techniques (a rate of $\frac{1}{\sqrt{n_{\text{ens}}}}$) and based on the similar performance for 10 and 20 ensemble members, a large increase in the ensemble size would be required to reach this goal. Because of computational limitations and since I expect little improvement in the mean result, I perform no further experiments for $n_{\text{ens}} > 20$.

While it may seem surprising that 10 ensemble members are sufficient to approximate the high-dimensional distribution of the model state. This is a result of the 2-dimensional error subspace that the particle filter operates in. Since state-perturbations are only caused by two parameters, it is sufficient that the ensemble can approximate the 2-dimensional parameter distribution and its time-evolution to obtain adequate results. An expanded state-augmentation that incorporates more than two parameters will likely profit from larger ensemble sizes.

4.5 Discussion

In this chapter, I applied the particle filter technique for data assimilation using a 3-dimensional biological ocean model and a set of daily satellite observations of chlorophyll for the purpose of joint state and parameter estimation. My goal was to assess particle filter application for a realistic model, large-scale, and the complexity of using real observations with significant outliers and anomalous values. To do this, I made use of an error subspace by allowing a small number (two) of the biological model parameters to be stochastic and so produce ensemble variation for the prediction step of the filter. I modified the sequential importance resampling (SIR) algorithm for particle filtering as follows. To increase the SIR's robustness with regard to outlying observations, I assimilated observations asynchronously and altered the observation update step by controlling the weight spreading that governs resampling of the ensemble. Finally, I used state-augmentation to allow for joint parameter and state estimation.

One of the main problems in using particle filter for data assimilation in realistic large scale scenarios is ensemble collapse and the accompanying low ensemble spread that prevents effective data assimilation. Ensemble collapse is often caused by an undersampling of the state space. A second cause is the number of independent observations, as large a number of successive assimilation steps increase the probability of ensemble collapse. Both issues have been reported for particle filter applications in high-dimensional state spaces (*Snyder et al.*, 2008; *van Leeuwen*, 2009). I have shown in this chapter, it can also be caused by outlying observations which are ubiquitous in biological applications. In high-dimensional applications, the state space is typically sparsely populated by a small ensemble relative to the number of dimensions. As a consequence, one or just a few ensemble members tend to lay close to the observation (in the sense of obtaining a high likelihood) and receive high weights in the particle filter-based assimilation procedure. This leads to the resampling of the few highly-weighted ensemble members and the collapse of the ensemble. A similar process can occur in the presence of outlying observations. Outliers deviate notably from the rest of the observations and tend therefore not to lay close to those regions of the state space where previous observations have consolidated the ensemble. Again, only the few close ensemble members receive high weights. As the experiment in Section 4.4.3 shows, outlying observations can bring an ensemble close to collapse and can therefore effectively impede data assimilation.

In this application, I avoid the problem posed by the high dimensionality of the model by effectively operating the data assimilation in a lower dimensional error subspace. This is achieved by introducing stochasticity into the ensemble solely by varying a few model parameters that the model is very sensitive to. Furthermore, the stochastic parameters are incorporated into the augmented state. The variation of parameters as a means by which stochastic variation is added to the ensemble, is especially suitable for biological models (*Annan*, 2001). These models contain parameters with a large range of possible values that are in practice not very well known and whose effect on the model output is great. For this chapter I chose parameters that the model chlorophyll is very sensitive to, ensuring adequate spread in the predictive ensemble with respect to chlorophyll for most of the simulation. At some points however, the predictive ensemble does not cover the observations well and as a consequence the improvement obtained by the particle filter assimilation remains relatively low (e.g. the surface chlorophyll development in November and December in the *open ocean* region in Figure 4.4).

I consider this work as a first assessment of the error subspace approach. With biological parameters only, model error, especially from physical sources (which play an important role, see the following chapter), is not well represented and other factors would have to be considered for a more comprehensive assessment (*Palmer et al.*, 2005). However, it is easy to add more and different error sources (either as augmented parameters or randomly drawn noise without memory) to the error subspace, likely at the cost of larger ensemble sizes. Additional work is required to assess the benefits and drawbacks of a higher dimensional error subspace and the effect of adding physical error sources on biological state and parameter estimation. Recently, some other particle filtering approaches suitable for state estimation for large scale systems have been proposed: *van Leeuwen* (2010) suggests carefully modifying the initial distribution of the error sources (in this application: θ_1 and θ_2) with the aim of preventing weight collapse by using future observational information, and *Chorin and Tu* (2009) suggest a nonlinear particle filter based on implicit sampling that behaves well for low numbers of particles. My use of error subspace provides an alternative approach, useful for both state and parameter estimation.

To address the detrimental effect of observational noise and outliers, I make use of ADA in my particle filter implementation. Without ADA, single anomalous observations can introduce strong shifts into the ensemble during assimilation (Figure 4.5(d)) whose effect persists over multiple assimilation steps and decreases the fit to the observations significantly. In fact, the particle filter configuration without ADA produces some of the worst results among my particle filter experiments (Table 4.3). The simple ADA scheme which I introduce in Section 4.3.4 is straightforward to implement and essentially averages over multiple observations before assimilating them into the ensemble. The second implementation aspect that increases the particle filter's robustness is the weighting during resampling. Here, I base the particle filter's weights on transformed inverse distance values, and introduce a parameter that allows for suitable spreading of the weights. This approach is flexible, as it allows the use of various types of observations along with suitable model-data distance measures.

An effect of using large-scale numerical models is that their computational cost effectively precludes the use of large ensemble sizes (even in very high dimensional problems, such as numerical weather prediction, ensemble sizes are between 5 and 100 (*Gneiting and Raftery*, 2005)). Ensemble-based data-assimilation techniques (e.g. particle filters) that are built on these relatively small samples can show strong variability in their results from one experiment to the next. To quantify this variability I ran multiple replicates of my main experiments (see Table 4.2). Most particle filter experiments stayed close to the reference simulation, yet I observed some variation, both beneficial and detrimental. The use of multiple realizations to quantify sampling variability is standard tool in statistics (*Gelman and Rubin*, 1992), and its use is important in assessing the effectiveness of all data assimilation schemes relying on small ensembles.

State-augmentation allows for joint state and parameter estimation within the particle filter procedure (*Dowd*, 2011). In my experiments, the particle filter benefits from this resampling of parameters contributing to consistent chlorophyll estimates and allowing parameters to diverge from ill-posed initial distributions. For the two biological parameters that are part of the state-augmentation it is realistic to assume time-dependence (see Chapter 3). Parameter estimates obtained via state-augmentation could reliably identify the low frequency time evolution of θ_1, the chlorophyll-to-carbon ratio (Figure 4.5(a) and (b)). For θ_2, the less sensitive zooplankton grazing parameter, the estimates are less reliable and vary greatly amongst different particle filter simulations. In this application, it appears that state-augmentation is more useful to obtain better state estimates than to obtain reliable parameter estimates.

4.6 Conclusions

I have implemented a particle filter algorithm for data assimilation of real daily satellite observations of chlorophyll into a realistic time dependent 3-dimensional biological ocean model. The particle filter offers advantages over other ensemble data assimilation methods in that it produces the correct target distributions for the state and parameters for general nonlinear models and non-Gaussian errors, a feature which is not supported by the widely used ensemble Kalman filter (*van Leeuwen*, 2009). I have provided modifications to the particle filter that allow it to work successfully with realistic, high-dimensional models and in the presence of outlying observations. This advancement offers the possibility for state-of-the-art, statistically rigorous sequential ensemble data assimilation to be applied to high-dimensional ocean models and observations.

CHAPTER 5

SENSITIVITY AND UNCERTAINTY ANALYSIS OF MODEL HYPOXIA ESTIMATES FOR THE TEXAS-LOUISIANA SHELF

5.1 Introduction

Quantification of uncertainty is an important part of numerical modelling. Knowledge of model uncertainty allows for an assessment of the reliability and precision of the model and therefore its general usefulness as a tool for prediction and analysis (*Karniadakis and Glimm*, 2006). Inherently connected to the concept of model uncertainty is that of model sensitivity. Model uncertainty is the model error which is due to lack of knowledge of the simulated system or due to lack of (computational) resources to simulate the degree of complexity of the system. Model sensitivity characterizes the response of the model output to changes in its input; a model is said to be sensitive to a particular input if a slight change in the input can trigger a large change in the model output. Because many model inputs are not well known and contain uncertainty, the sensitivity of an input determines the effect of input uncertainty on model uncertainty.

In this chapter, I perform a combined sensitivity and uncertainty analysis for a complex 3-dimensional physical-biological ocean model. The model simulates nitrogen-limited primary production and dissolved oxygen on the Texas-Louisiana shelf in the northern Gulf of Mexico. It is aimed at estimating the occurrence and extent of hypoxia, defined as the oxygen concentrations below a critical threshold. The goal of this chapter is to assess the sensitivity of biological model output and hypoxia estimates to example inputs of various categories: on the biological side of the model, my analysis includes one example parameter that influences the nitrogen cycle, biological boundary and initial conditions, and nutrient river input. For the physical model, I assess the impact of the parametrization of sub-grid scale horizontal mixing, the wind input as an example of physical forcing with significant impact on the extent of hypoxia (*Feng et al.*, 2012), and freshwater river input. From the results, I gain an understanding of the relative importance of the inputs on biological properties and the temporal and spatial distribution of model uncertainty.

In oceanography, previous studies of model uncertainty and sensitivity are typically applied to physical ocean models (*Lermusiaux*, 2006; *Kim et al.*, 2010; *Thacker et al.*, 2012). Sensitivity analyzes are also commonly found in the field of ecosystem modelling (*Clancy et al.*, 2010; *Makler-Pick et al.*, 2011; *Melbourne-Thomas et al.*, 2011) analysing models that are typically not based on 3-dimensional physical models. Only few studies have investigated uncertainty propagation in physical-biological ocean models: *Béal et al.* (2010) assess the effect of mixing errors on biological properties in a physical-biological model.

Unless uncertainty is directly integrated into the model (see e.g. *Lermusiaux*, 2006) an uncertainty or sensitivity analysis entails many model simulations and is thus computationally demanding, in particular for complex models. Typically, uncertainty analyses represent input uncertainty via random samples using Monte Carlo techniques (*Clancy et al.*, 2010; *Kim et al.*, 2010; *Melbourne-Thomas et al.*, 2011), a computationally inefficient approach that may not be practicable for complex models. Emulator-based approaches, such as the polynomial chaos expansion (introduced by *Wiener*, 1938), offer a computationally more efficient alternative for the task of propagating uncertainty in model inputs to the output (*Xiu and Karniadakis*, 2003; *Shen et al.*, 2010; *Thacker et al.*, 2012). They propagate uncertainty by sampling the input distribution in a non-random fashion and interpolating model output in between samples. For this sensitivity and uncertainty analysis, I use the polynomial chaos expansion and assess its use for sensitivity analyzes for biological ocean models.

5.2 Methods

5.2.1 Uncertainty Propagation with the Polynomial Chaos Expansion

Emulators are surrogates for models that approximate unknown model output based on existing output. Emulators offer a suitable way to propagate uncertainty in model inputs (the model's parameters, initial or boundary conditions, physical forcing, etc.) to its outputs (*Xiu and Karniadakis*, 2003; *Matre et al.*, 2004). For this purpose, emulators require output from model simulations that are based on variations of the uncertain inputs. By interpolating between the existing output values, emulators can estimate model output for new, unknown values of the inputs and even approximate the probability distribution of the model output. The polynomial chaos expansion, the emulator technique that is used in this chapter, is based on polynomial interpolation. In Chapter 3, I focused on the interpolation feature of polynomial chaos to estimate parameter values for a biological ocean model. In this chapter I focus on the polynomial chaos-based approximation of the model output probability distribution and the estimation of some of its properties.

I provided an introduction to the polynomial chaos expansion in Chapter 3, here I briefly review the main aspects with regard to uncertainty propagation. In the following I consider the stochastic model input θ, which can be, for example, a model parameter or a scaling factor for the model's boundary condition. Because I only consider variations of individual (and 1-dimensional) inputs in this chapter, I assume in the following that θ is 1-dimensional. Although I do not consider the multi-dimensional case here, the theory translates in a straightforward manner into two or more dimensions for multiple, independent inputs (*Xiu and Karniadakis*, 2002).

Here, the model output of interest is represented by the function $f(\mathbf{x}, t, \theta)$. The output may be dependent on space $\mathbf{x}$, time t and the uncertain input θ. In the polynomial chaos expansion, f is approximated by a basis function expansion:

$$f(\mathbf{x}, t, \theta) = \sum_{k=0}^{k_{\text{max}}} a_k(\mathbf{x}, t)\ \phi_k(\theta) \quad + \epsilon_{\text{trunc}}(\theta) \tag{5.1}$$

where $a_k(\mathbf{x}, t)$ are expansion coefficients, independent of the uncertain input θ, and the kth basis function $\phi_k(\theta)$ is a polynomial of order k in the parameter space defined by θ. The parameter k_{max} is the maximum order of polynomials used in the approximation and determines the quality of the approximation and ϵ_{trunc} is the truncation error. Without cutoff, i.e. for $k_{\text{max}} = \infty$, the approximation is exact and $\epsilon_{\text{trunc}}(\theta) = 0$. However, the number of required model runs grows with k_{max}, so that computational constraints force me to use relatively small values in typical applications using complex models.

The choice of polynomials in equation (5.1) is dependent on the probability density function (pdf) of the parameter θ which I denote $p(\theta)$. The polynomials are chosen to be orthogonal with respect to p and all common distributions have specific sets of polynomial basis functions (*Xiu and Karniadakis*, 2002). For the beta distribution, which is used here for its finite support, the corresponding set of orthogonal polynomials are the Jacobi polynomials and ϕ_k is the kth Jacobi polynomial.

To approximate f in Equation (5.1), the coefficients a_k need to be computed which in turn requires the computation of $f(\mathbf{x}, t, \theta^{(i)})$ at each quadrature point $\theta^{(i)}$ for $i = 0, 1, \ldots, k_{\text{max}}$. These are the points where existing model output of $k_{\text{max}} + 1$ model simulations is required for the emulator-based approximation. Once the coefficients have been computed, two important properties of the output distribution, the expected value and variance of f are straightforward to calculate. They are given by

$$\mathbb{E}(f(\mathbf{x}, t, \theta)) = a_0(\mathbf{x}, t) \tag{5.2}$$

$$\text{var}(f(\mathbf{x}, t, \theta)) = \sum_{k=1}^{n} a_k^2(\mathbf{x}, t)\ N_k. \tag{5.3}$$

Here $N_k = \int_S \phi_k(\theta)^2\ p(\theta)\ \mathrm{d}\theta$ is a normalization factor specific to the kth polynomial and independent of θ; S is the support of p (the region where $p(\theta) > 0$). The properties in Equations (5.2) and (5.3) are the mean and variance of the probability distribution of f based on the distribution of θ. Different inputs will produce different distributions in f, therefore mean and variance are dependent on θ.

When the polynomial chaos-based approximation is expanded beyond a single (1-dimensional) input, the computational cost for the standard polynomial chaos expansion increases exponentially with the number of stochastic inputs (and the number of dimensions of each input). As a result, a joint analysis of 8 inputs (each 1-dimensional) with polynomials of order k_{max} would require $(k_{\text{max}} + 1)^8$ model runs, while only $(k_{\text{max}} + 1) \times 8$ are required for an individual analysis of each input. Although methods have been proposed to reduce the number of required simulations by performing sparse sampling (see e.g.

Blatman and Sudret, 2010), I am specifically interested in the effect of individual inputs on the model under similar conditions. For this reason, I sample each input distribution in the same manner but restrict the uncertainty analysis here to one parameter at the time.

5.2.2 Model Description

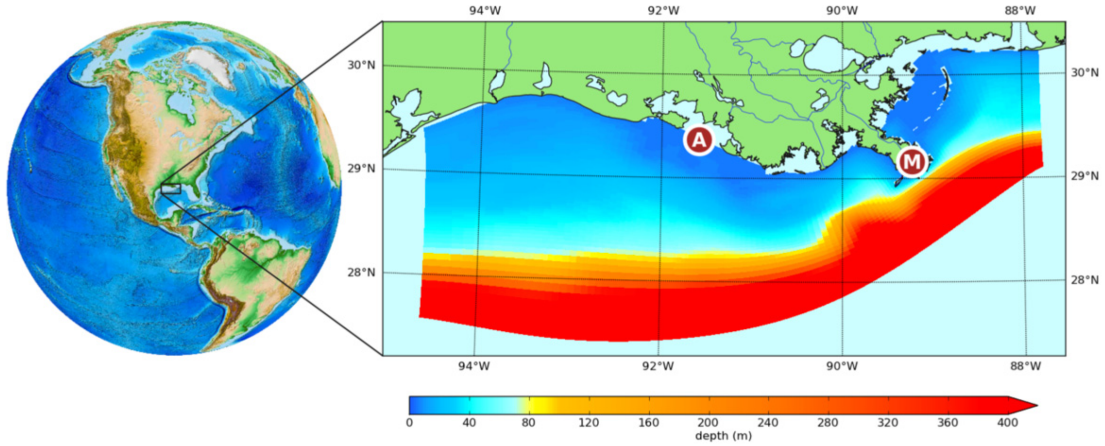


Figure 5.1: The model domain and its bathymetry. The Mississippi delta and the Atchafalaya mouth are marked by brown circles with “M” and “A”, respectively.

The physical-biological ocean model of the Texas-Louisiana shelf in the northern Gulf of Mexico (Figure 5.1) is based on a configuration of the Regional Ocean Modeling System (ROMS; *Haidvogel et al.* (2008)) with a biological component (*Fennel et al.*, 2006). The model is taken directly from *Fennel et al.* (2011) where it is described in detail. Here I provide a short overview with a focus on the model inputs with uncertainty. The main objective of the model is to simulate the biological nitrogen cycle and the distribution of disolved oxygen in the shelf region which is heavily influnced by runoff from the Mississippi and Atchafalaya rivers *Bianchi et al.* (2010). For this reason the model domain is centered around the mouths of these rivers and stretches along the coast, in the region that is most heavily influenced by river runoff (Figure 5.1).

5.2.2.1 Hypoxia on the Texas-Louisiana Shelf

The Mississippi-Atchafalaya river system is fed by a vast 3 220 000 km^2 drainage basin and supplies a high load of nutrients (recent estimates are at around 1.25 Tg nitrogen per year *Aulenbach et al.*, 2007) to the Texas-Louisiana shelf. The nutrients fuel phytoplankton growth on the shelf and contribute to the formation of zones of low dissolved oxygen in bottom waters. Oxygen is depleted when sinking organic material is remineralized by bacteria consuming oxygen in the process. In summer, when the water is highly stratified and the ocean bottom is cut off from the oxygen resupply from the air, zones of low oxygen form. These zones are called hypoxic if the oxygen concentration is below a critical threshold of 63 mmol O_2 m^{-3} (the equivalent of 2 mg L^{-1}) that is considered harmful to many marine organisms. In fall, when the water column becomes less stratified the hypoxic area begins to shrink and disappear before reforming the following year. Hypoxia on the Texas-Louisiana shelf has been subject to many studies, a recent overview is given by *Bianchi et al.* (2010).

5.2.2.2 Instant Remineralization (IR) and Sediment Oxygen Demand (SOD) Model Configurations

Biochemical processes that take place in the sediments have a major influence on the formation and development of hypoxic zones. Some of the sinking organic material reaches the ocean floor and is slowly remineralized in the sediments. The model formulation offers two simplified ways to mimic the biochemical processes in the sediments: instant remineralization (IR) and sediment oxygen demand (SOD) based on a physical parametrization. As the name suggests, sinking organic material is remineralized instantaneously when it reaches the ocean bottom in the IR configuration. This approach has the advantage of making the oxygen consumption dependent directly on the amount of organic matter that reaches the sediment, yet has the disadvantage of speeding up the process of remineralization which in reality leads to a slower but longer lasting consumption of oxygen. In the SOD configuration, sediment oxygen consumption and nutrient production are parametrized and dependent on temperature and assumed to occur only in water depths shallower than 50 m water depth where detritivores are present. This approach is able to simulate a more steady, slow consumption of oxygen, yet the oxygen flux is independent of the flux of organic matter to the bottom and decoupled from the supply of nutrients. Here, I examine both the IR and the SOD model configurations and their response to uncertainty in the inputs. Both configurations use negative oxygen concentrations as a simplified way to account for oxygen dept due to the accumulation of reduced species under anaerobic conditions.

5.2.3 Model Inputs of Interest

In my experiments I introduce uncertainty to the model inputs listed in Table 5.1. All

name	abbreviation[a]	description
river runoff	*river*	Variation in the freshwater river runoff including all chemical and biological constituents such as nutrients (NO_3 and NH_4) and detritus, which are held at constant concentrations. The variations are introduced to both the Mississippi and the Atchafalaya river.
freshwater river runoff	*disch*	Variation in the river runoff excluding the chemical and biological constituents which are held at constant loads. Here, variations are introduced in the fresh water discharge of the Mississippi and the Atchafalaya River only.
phytoplankton maximum growth rate	μ_0	Variation in the temperature and nutrient dependent growth rate of phytoplankton, one of the parameters governing the nitrogen cycle in the biological model.
nutrient initial conditions	*nut ini*	Variation in the initial conditions for the two biological nutrient variables NO_3 and NH_4.
nutrient boundary conditions	*nut bry*	Variation in the boundary conditions for the two biological nutrient variables NO_3 and NH_4.
horizontal diffusivity	*hdiff*	Variation in the horizontal mixing coefficient for tracer variables.[b]
horizontal viscosity	*hvisc*	Variation in the horizontal mixing coefficient for momentum.[b]
wind	*wind*	Variation in wind speed over the entire model domain.

[a] used in figures

[b] The horizontal mixing coefficients provide a parametrization of sub-grid scale horizontal mixing.

Table 5.1: The model inputs that are varied in the uncertainty experiments in this chapter.

variations I introduce into the model input are simple scalings. For example, uncertainty is introduced into the wind field by scaling the wind speed up and down across the entire domain. The univariate scaling factor can be approximated at a much lower computational cost than a multidimensional distribution which would be required to represent temporal or spatial variations in the input uncertainty. An alternative approach to vary multidimensional input is a dimension reduction, e.g. via principal component analysis (see *Thacker et al.*, 2012), and subsequent variation of the lower dimensional representation.

5.2.3.1 Input Distributions

My aim is to quantify and compare the effect of uncertainty in various model inputs on the model output. For this purpose, I apply the same relative amount of variation to each of the inputs. That is, I use a distribution with a constant ratio of standard deviation/mean for all inputs. While these input uncertainties do not reflect my best knowledge of the real uncertainty of the inputs (accurate uncertainty estimates would be hard to obtain for most of the inputs), this approach allows for straightforward comparison of the effect on the output uncertainty, in keeping with sensitivity analyzes.

All variations in the input are based on the variation of a scaling factor that is multiplied with the input quantity, e.g. the amount of runoff from the rivers in the model. For a scaling factor that is equal to one, I obtain the baseline model simulation with standard parameters, which represents the scenario I consider the most likely. Departure of the scaling factor from 1 will create perturbations in the model.

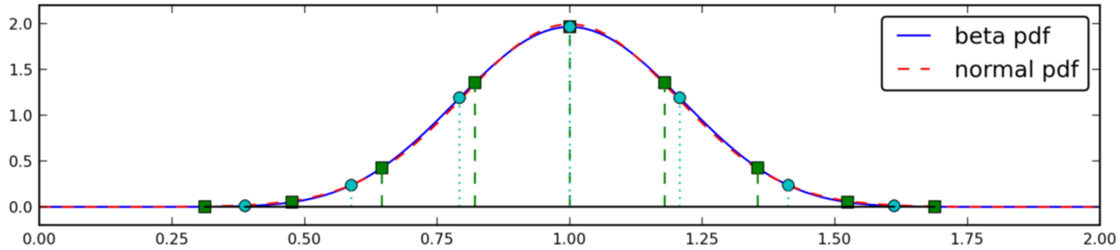


Figure 5.2: The pdf of the scaled beta distribution in a comparison to a normal distribution with equal mean (1.0) and standard deviation (0.2). The beta pdf is zero beyond the $[0, 2]$ interval, whereas the support of the normal distribution is not limited. Cyan circles and green squares mark the 7 and 9 polynomial chaos quadrature points, respectively, at which the pdf is sampled in the polynomial chaos expansion.

For my experiments, I assume a beta distribution with a scaled support, so that it allows for values of the scaling factor between 0 and 2. The scaled beta distribution has the probability density function

$$p(x) = \frac{\frac{1}{2}(\frac{1}{2}x)^{\alpha-1}\,(1-\frac{1}{2}x)^{\beta-1}}{\mathrm{B}(\alpha,\beta)} \text{ for } x \in]0,2[. \tag{5.4}$$

Here, α and β are positive parameters that determine the shape of p, and $\mathrm{B}(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}\mathrm{d}x$ is the beta function which acts as a normalizing constant. I only consider beta distributions with parameters $\alpha = \beta > 1$ which results in symmetric distributions with mean values of 1 and with variances that can be adjusted via α and β. Here, I select a standard deviation of 0.2, the resulting distribution is similar to a Normal distribution with equal mean and standard deviation (see Figure 5.2). The advantages of the beta distribution is that it is truncated and thus does not allow for scaling factors less than 0. The beta distribution also comes with its associated set of orthogonal polynomials, the *Jacobi* polynomials (*Xiu and Karniadakis*, 2003), simplifying the implementation of the polynomial chaos expansion.

5.2.4 Quantifying Uncertainty in Hypoxic Area Estimates Using Shannon Entropy

A parcel of water is defined as hypoxic if its dissolved oxygen concentration is below the critical threshold of 63 mmol O_2 m^{-3}. In the model, the hypoxic area comprises all grid cells with an oxygen concentration below the threshold. If I consider a single grid cell at a fixed point in time, it can be in one of two states, it can either be hypoxic or not. When uncertainty is introduced into the model, the grid cell has a certain probability of being hypoxic. Now the aim is to identify a suitable measure to quantify the uncertainty in the state of the grid cell. If the probability of being hypoxic is zero, it is certain that the grid cell is not hypoxic and the measure of uncertainty should be 0 to indicate that there is no uncertainty. For the same reason the measure of uncertainty should be 0 if it is certain that hypoxia is occurring and the probability of being hypoxic is one. In all other cases the uncertainty measure should be greater than zero and should reach its maximum when both hypoxia and no hypoxia have equal probability, i.e. when the probability of hypoxia is $\frac{1}{2}$.

A measure of uncertainty that fulfils the above criteria is the *Shannon entropy* introduced in *Shannon* (2001) (originally published in 1948). It is defined for a discrete random variable X that has n possible values $x_1, x_2, \ldots, x_n$ as

$$H(X) = -\sum_{i=1}^{n} p_i \, \log_b(p_i). \tag{5.5}$$

Here $p_i = \text{Prob}(X = x_i)$ is the probability of X taking the value x_i, and b is the base of the logarithm that can be chosen on an application dependent basis. For any of the $p_i = 0$, the product $p_i \, \log_b(p_i) = 0$ as well.

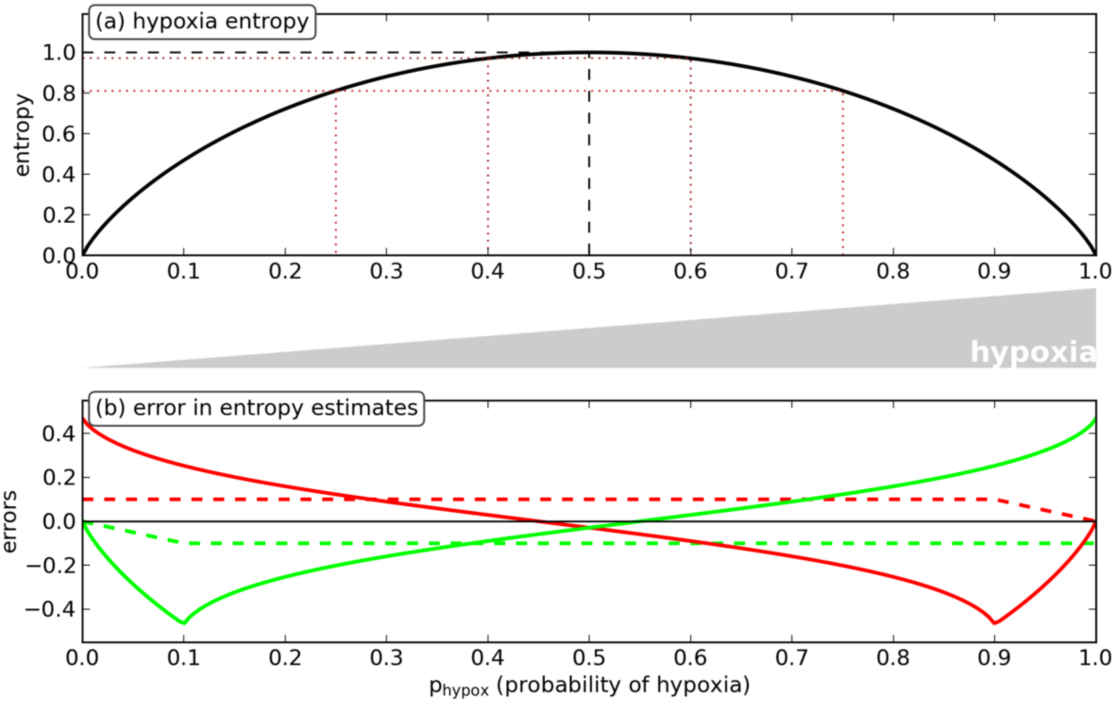


Figure 5.3: The relationship of p_{hypox}, the probability of hypoxia, and hypoxia entropy (a) and the relationship between error in p_{hypox} and error in hypoxia entropy (b). The dashed lines in (b) depict an error of 0.1 in p_{hypox} in form of an underestimation (green) and overestimation (red; errors below 0.1 are due to restricting p_{hypox} to values between 0 and 1). The solid lines show the resulting error in the entropy propagated via the function in (a) (see Equation (5.6)).

In this application, X is the state of a grid cell which can take one of $n = 2$ values: $x_1 =$ "hypoxia" and $x_2 =$ "no hypoxia". I let $p_{\text{hypox}} = p_1$ be the probability of X being hypoxic and choose $b = 2$. Then the Shannon entropy simplifies to

$$H(X) = -p_{\text{hypox}} \, \log_2(p_{\text{hypox}}) \; - \; (1 - p_{\text{hypox}}) \, \log_2(1 - p_{\text{hypox}}). \tag{5.6}$$

The above expression is shown in Figure 5.3(a) as a function of p_{hypox}, it behaves as desired: If the probability of hypoxia is zero ($p_{\text{hypox}} = 0$) or one ($p_{\text{hypox}} = 1$) the Shannon entropy is zero. In between, the Shannon entropy is greater than zero and at $p_{\text{hypox}} = \frac{1}{2}$ it reaches its maximum value of one. These properties indicate intuitively that the Shannon entropy is a suitable measure of uncertainty for hypoxic area estimates and one can show generally that the Shannon entropy fulfils the criteria expected from a measure of uncertainty for a discrete random variable (*Shannon*, 2001; *Jaynes*, 1957). In this chapter I will only use the Shannon entropy in its binary form (5.6) and refer to it simply as *entropy*.

5.2.5 Estimating Properties of the Output Distribution

After performing the required model simulations (performed for the input values at the quadrature points; see Section 5.2.1), I obtain the model output at the quadrature points. These represent samples from the probability distribution of the model output. My goal is to estimate properties of the distribution from these samples. The polynomial chaos expansion offers a straightforward and numerically efficient way to estimate the mean and standard deviation of the output distribution (see Equations (5.2) and (5.3)). However, other properties I am interested in, such as the median, other quantiles and the entropy cannot be obtained that way. I need to resort to a procedure of interpolation and binning for their approximation.

In the following, I consider a 0-dimensional (point) model output $\mathrm{f}(\theta)$ that is dependent on the 1-dimensional model input θ. The procedure below describes the approximation of $p_{\mathrm{f}}(\mathrm{f}(\theta))$, the pdf of the model output, based on $p(\theta)$, the pdf of θ. For higher dimensional model output the procedure needs to be repeated for every entry in the output vector or matrix:

(i) Interpolate the model output in parameter space, i.e. estimate $\hat{\mathrm{f}}(\theta) \approx \mathrm{f}(\theta)$ based on the model output at the quadrature points. Any suitable interpolation technique can be used here, in this chapter polynomial interpolation based on the polynomial chaos expansion is a natural choice.

(ii) Create a fine equidistant grid in parameter space, covering values of θ where $p(\theta)$ is (significantly[1]) greater than zero. Let θ_i^{grid} for $i = 1, 2, \ldots, n$ denote the resulting n grid points.

[1] If $p(\theta)$ is greater than zero on an unlimited interval, e.g. in case of a normal distribution, it is sufficient to only include those regions of the parameter space where $p(\theta)$ is greater than a small, positive threshold.

(iii) Obtain values of $\hat{\mathrm{f}}$ and p for each grid value, i.e. obtain pairs $\left(\hat{\mathrm{f}}(\theta_i^{\mathrm{grid}}), p(\theta_i^{\mathrm{grid}})\right)$ for $i = 1, 2, \ldots, n$.

(iv) Bin the values of $\hat{\mathrm{f}}(\theta_i^{\mathrm{grid}})$ for $i = 1, 2, \ldots, n$ into m bins (non-overlapping intervals). Let x_j for $j = 1, 2, \ldots, m+1$ be the limits of the bins.

(v) Sum the probability values associated with each bin, i.e. compute

$$\hat{s}_j = \sum_{i\,:\,x_j \leq \hat{\mathrm{f}}(\theta_i^{\mathrm{grid}}) < x_{j+1}} p(\theta_i^{\mathrm{grid}}) \qquad \text{for } j = 1, 2, \ldots, m.$$

(vi) Finally, normalize the sums to obtain $s_j = \frac{\hat{s}_j}{\sum_{k=1}^{m} \hat{s}_k}$, which is an approximation of the probability of the model output being in the jth bin, i.e. $s_j \approx \int_{x_j}^{x_{j+1}} p_{\mathrm{f}}(x)\mathrm{d}x$.

At this point I have associated model output values with probability values. After the normalization in step (vi) I obtain an estimated pdf for the output. This pdf can then be used to obtain the desired distribution properties. In practice, I use 100 bins of equal size to cover the interval between minimum and maximum value of the interpolated model output (step (iv)) in conjunction with 1000 grid points (step ii). An increase in the number of bins typically requires a finer grid with more points.

For the purpose of estimating the entropy of a binary event (such a having model oxygen output less than the hypoxic threshold; see Section 5.2.4), it is sufficient to use two bins in step (iv) so that all output values below the threshold are collected in the first bin. After the normalization step (vi), the sum of input pdf values in the first bin is an estimate of the probability of the event having the first of two possible outcomes (in my example: an oxygen concentration below the threshold).

I tested the above procedure by estimating the mean and standard deviation of multiple output distributions and comparing the values with the estimates obtained directly from the polynomial chaos expansion. Both estimates differ by at most 0.005% in my tests and thus agree to the precision that is relevant here. Note, that the above procedure can be replaced by a Monte Carlo procedure based on random samples from the input distribution. However, I found the Monte Carlo procedure to be less efficient as it required a larger sample size and thus more computing time. No matter which of the aforementioned procedures is used to turn the model output into estimates of the model distribution, the limiting factor for their accuracy is typically the number of quadrature points. More quadrature points lead to a smaller error in the interpolation and more accurate estimates, but they also require more model simulations (see Section 5.2.1).

5.2.6 Uncertainty in Emulator Estimates

The emulator approximation of model output contains the truncation or interpolation error $\epsilon_{\text{trunc}}(\theta)$ (see Equation (5.1)). To estimate the magnitude of this error for the biological output and the derived uncertainty estimates (see Section 5.2.5), I compare the emulator estimates for two different numbers of quadrature points, i.e. for two different values of k_{max}. Based on the experiments in Chapter 3 with 7 quadrature points, I select $k_{\text{max}} = 6$ and 8 which corresponds to emulator approximations based on 7 and 9 quadrature points, respectively. For both values of k_{max} I obtained the emulator approximations for most model inputs[2] (see Section 5.2.3).

If the model is well-approximated by the emulator, then the change from 7 to 9 quadrature points should be accompanied by a small reduction in the truncation error and little change in the output estimates. If, on the contrary, the output estimates vary considerably, this indicates that the output is not approximated well by the emulator with 7 quadrature points (indeed, a large change in $\epsilon_{\text{trunc}}(\theta)$ would indicate that the sum in Equation (5.1) is not close to converging and thus even 9 quadrature points are not sufficient).

By comparing selected uncertainty estimates that are based on the emulator approximations for 7 and 9 quadrature points, I found that the truncation error varied considerably among different inputs. Based on the magnitude of the error, I can divide the inputs into two distinct classes. The first class consists of inputs that induce changes into the biological model directly and do not affect the physical model (e.g. biological parameters, biological initial and boundary conditions). The effect of these inputs is well-approximated by the emulator and the 7 and 9 point-based uncertainty estimates are very similar (see Figure 5.4). The second class of inputs consists of those that change the physics of the model and thereby indirectly alter the biology (e.g. both river runoff scenarios, diffusivity, and wind). The biological model output appears to be very sensitive to changes in these parameters and the emulator-based approximation is not precise in places and can vary greatly between 7 and 9 quadrature points (Figure 5.4).

[2] I did not obtain approximations for the *nutrient initial conditions* and *nutrient boundary conditions* inputs based on 9 quadrature points. These inputs have such a low effect on the output (see e.g. results in Figure 5.6) that I feel confident that 7 quadrature points are sufficient for the emulator approximation.

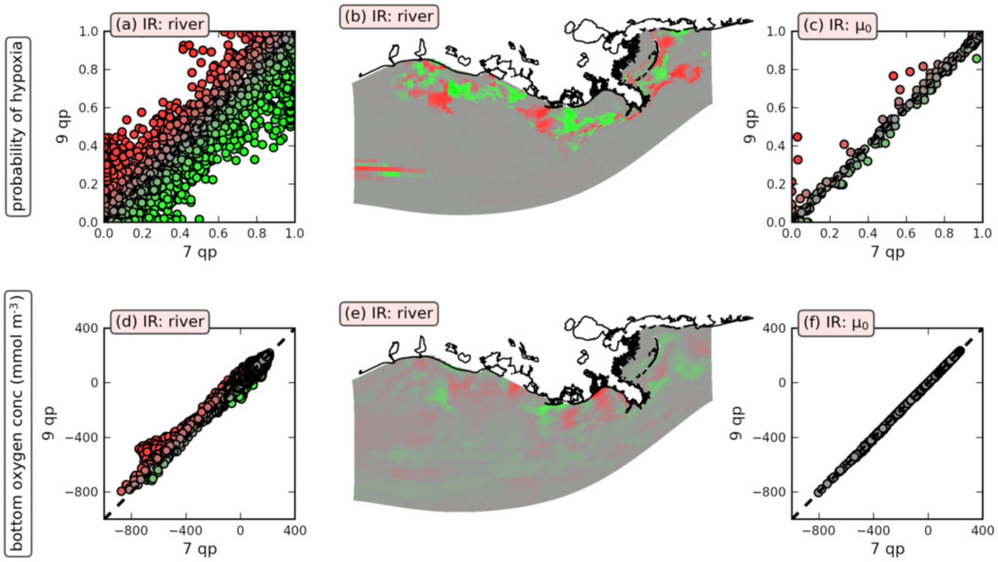


Figure 5.4: Comparison of emulator-based estimates of the probability of bottom hypoxia (top row; (a)-(c)) and the bottom oxygen concentration (bottom row; (d)-(f)) for 7 and 9 quadrature points (qp) at a day in mid July for instant remineralization (IR) configuration. The scatter plots depict the variation in the emulator estimates for *river*, one of the physical inputs, and μ_0, a biological input. In the scatter plots, distance from the diagonal is emphasized by colour intensity. For the physical input, which displays a higher variation, the maps ((b) and (e)) visualize the spatial location of the values with high variation (the colours correspond to those in the scatter plot (a) and (d), respectively).

To explore if the truncation error exhibits any dependence on space, I map the error values back onto the model grid (Figure 5.4(b) and (e)). The spatial dependence of uncertainty in both estimated mean chlorophyll concentrations and estimated bottom hypoxia probabilities is striking. High uncertainties in the emulator estimates are constrained to clusters along the coast and are especially evident close to the location of the Mississippi and Atchafalaya mouths. My results in Section 5.3 show that these regions of elevated emulator uncertainty are highly dynamic regions that also feature the highest model uncertainty.

5.2.6.1 Propagation of Uncertainty into Entropy Estimates

In light of the high uncertainty for some of the emulator-based estimates of the probability of hypoxia (Figure 5.4), it is important to note how the uncertainty propagates from probability estimates to entropy estimates (see Section 5.2.4). Based on Equation (5.6), it is straightforward to compute the uncertainty in entropy based on the uncertainty in the probability estimates; due to the log function, the transformation is not constant (Figure 5.3). Instead, the uncertainty is increased for probability estimates close to zero or close to one (corresponding to low entropy values) whereas uncertainty is diminished at probability values close to 0.5 (corresponding to high entropy values). Low entropy values therefore have a higher uncertainty than high values.

5.3 Results

5.3.1 Baseline Oxygen Dynamics without Uncertainty

To gain an understanding of model dynamics prior to the introduction of uncertainty I evaluate the output of the baseline model simulations for the SOD and the IR configurations. These simulations use the standard model inputs without uncertainty and correspond to the model simulations with all scale factors set to one (see Section 5.2.3).

For the model evaluation, I focus on the temporal development of dissolved oxygen at one station on the shelf between the Mississippi Delta and the mouth of Atchafalaya Bay, from here on simply referred to as the *station* (see star in Figure 5.5(a)). To get a spatial overview, I examine a spatial snapshot of bottom oxygen across the model domain on 17 July 2004, a time when the hypoxic zone in the model has reached one of its largest extents. Both temporal and spatial oxygen fields are displayed in Figure 5.5 for the SOD and IR model configurations.

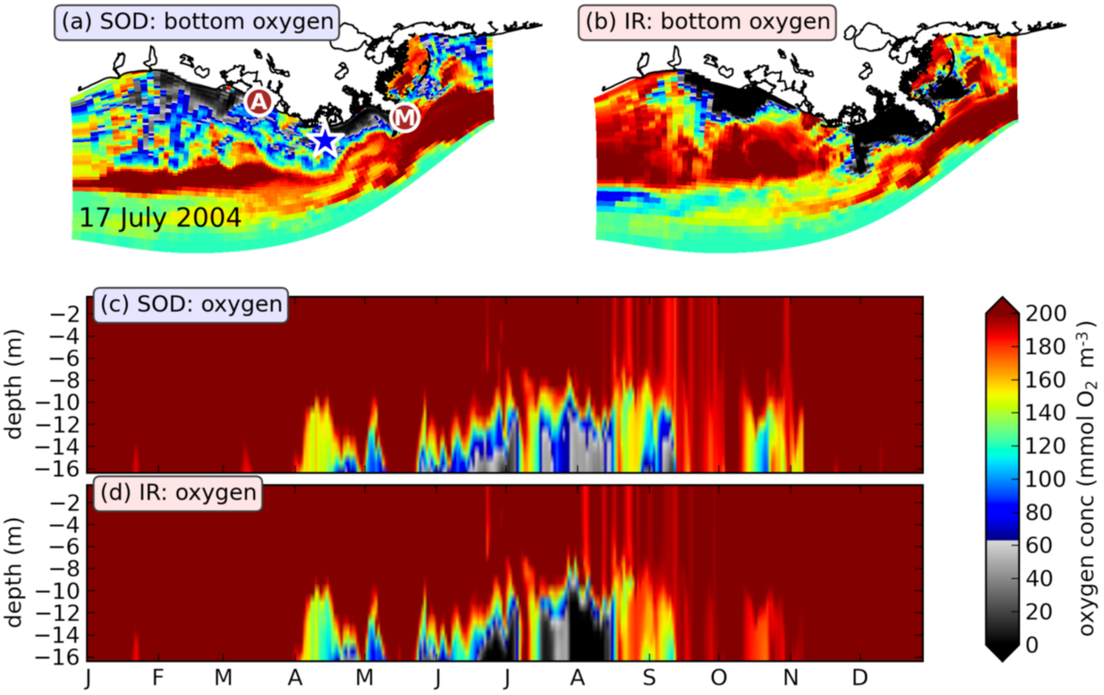


Figure 5.5: Snapshots of the simulated bottom oxygen concentration on 17 July 2004 for the SOD (a) and IR (b) configurations and corresponding time-depth plots of the oxygen development ((c) and (d)) at the station marked by the blue star in between the Mississippi (brown circle with "M") and the Atchafalaya (brown circle with "A") mouths. Grey and black colours mark hypoxic areas with oxygen concentrations below 63 mmol O_2m^{-3}.

The oxygen concentration at the location of the station exhibits a slow decline at the ocean bottom throughout the spring months of 2004, beginning in April. The decline is briefly interrupted by several mixing events that deliver oxygen-rich surface water to the bottom. Both SOD and IR models show hypoxic conditions at the sea floor in late May and strongly hypoxic conditions throughout most of July and the first half of August. In mid-August strong mixing increases bottom oxygen and leads to non-hypoxic conditions. The IR configuration exhibits more persistent hypoxic conditions with lower oxygen levels in comparison to the SOD model which switches more frequently between hypoxic and non-hypoxic conditions.

The spatial view (Figure 5.5(a) and (b)) reveals larger hypoxic areas along the coast close to the mouths of the Mississippi and Atchafalaya in both model configurations. Bottom oxygen concentrations increase farther away from the rivers and are especially high close to the shelf break. The station is at the fringe of the strongly hypoxic region that is fed by the Mississippi River. In the SOD configuration, the hypoxic area appears less homogeneous and more fractured in comparison to the IR configuration and the SOD bottom oxygen concentration field shows more small scale variability.

5.3.2 Uncertainty in Bottom Oxygen

I now introduce uncertainty into the model by means of varying the model inputs (see Section 5.2.3). First, I focus on the effect of the bottom oxygen concentration at the location of the station. To characterize the probability distribution of bottom oxygen and its seasonal development, I compute its quantiles averaged over periods of 3 months (see Fig 5.6). Two main results are immediately apparent in the SOD and IR model configurations when comparing the quantiles. Based on the interquartile range (the difference between the 0.75 and 0.25 quantiles) which characterizes the dispersion of the output distribution, the effect of the different inputs varies considerably and their effect shows a strong seasonal dependence.

Uncertainties in the wind input, the main driver of vertical mixing, have the strongest effect on bottom oxygen for both model configurations. Effects of similar magnitude are caused by the horizontal mixing coefficients (*hdiff*, *hvisc*) and both river inputs (*river*, *disch*). On the other side of the spectrum, nutrient initial and boundary conditions have little to no impact on the bottom oxygen concentration at the station. The phytoplankton growth rate (μ_0) is the only input exhibiting a distinctly different effect on the two model

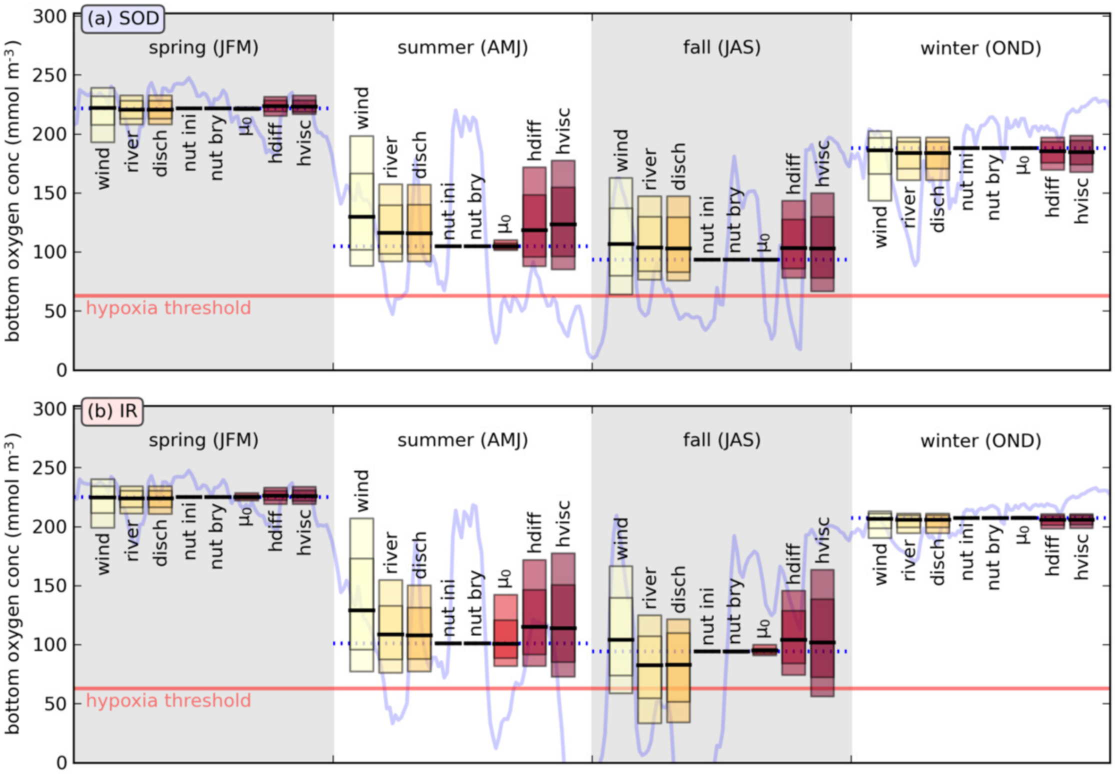


Figure 5.6: Seasonal (time-averaged) quantiles of the bottom oxygen distribution at the station marked in Figure 5.5 for 2004. Each bar marks the region between 0.1 and 0.9 quantile (outer bar), the region between the 0.25 and 0.75 quantile (inner bar), and the median (solid black line) for one uncertain input, in one season in 2004. The blue solid line in the background marks the development of bottom oxygen in the baseline simulation without uncertainty; its seasonal averages are displayed as dotted blue lines.

configurations. In the IR configuration where sediment oxygen consumption is determined by sinking organic material, the effect of μ_0 is relatively strong, compared to the SOD configuration where oxygen consumption is decoupled from the biological nutrient cycle. One further notable point is the very similar effect of the *river* and *disch* input in the IR configuration; the additional variation of the river nutrient supply appears to have little added effect. Some of the distributions exhibit considerable skewness and a deviation of the median from baseline (the no uncertainty mean; dotted blue line in Fig 5.6), which are due to the nonlinearity of the model.

The time dependence of the effect of the uncertain inputs is consistent across inputs for both model configurations. In the middle of the year (April through September) the effect of the inputs on bottom oxygen is considerably higher than during rest of the year. The reason for the low effect early and late in the year is that temperatures are lower and the ocean is well-mixed, model behaviour that cannot be disrupted by the uncertainty in the input I prescribed.

5.3.3 Entropy of Hypoxia

Temporal Development of Entropy

Now I focus on the entropy of hypoxia which quantifies the uncertainty in model hypoxia estimates which reaches its highest value if the probability of hypoxia is 0.5 (see Section 5.2.4). The temporal development of hypoxia (Figure 5.7) shows low entropy values for most of the year and near the ocean surface. These values correspond to a low uncertainty due to a low probability of hypoxia; variation of the model inputs has a very low probability of causing hypoxia near the surface, or early or late in the year. High entropy values and therefore high uncertainty can only be found close to the ocean bottom and from mid-May to mid-September, in the times and places where hypoxia is likely to be induced. Here, the SOD and IR model configurations show some differences.

In the SOD configuration, the inputs that have a strong effect on bottom oxygen (see previous Section 5.3.1) cause high uncertainties in bottom hypoxia from June through September. The maximum likelihood region of hypoxia, where the probability of hypoxia is greater than 0.5 (region outlined in green in Figure 5.7), varies strongly among different model inputs (compare Figure 5.7(a)-(d)) and is characterized by high entropy values. Thus estimates concerning the number and duration of hypoxic events that the station experienced in 2004 are highly uncertain. This, of course, is only true for the model inputs

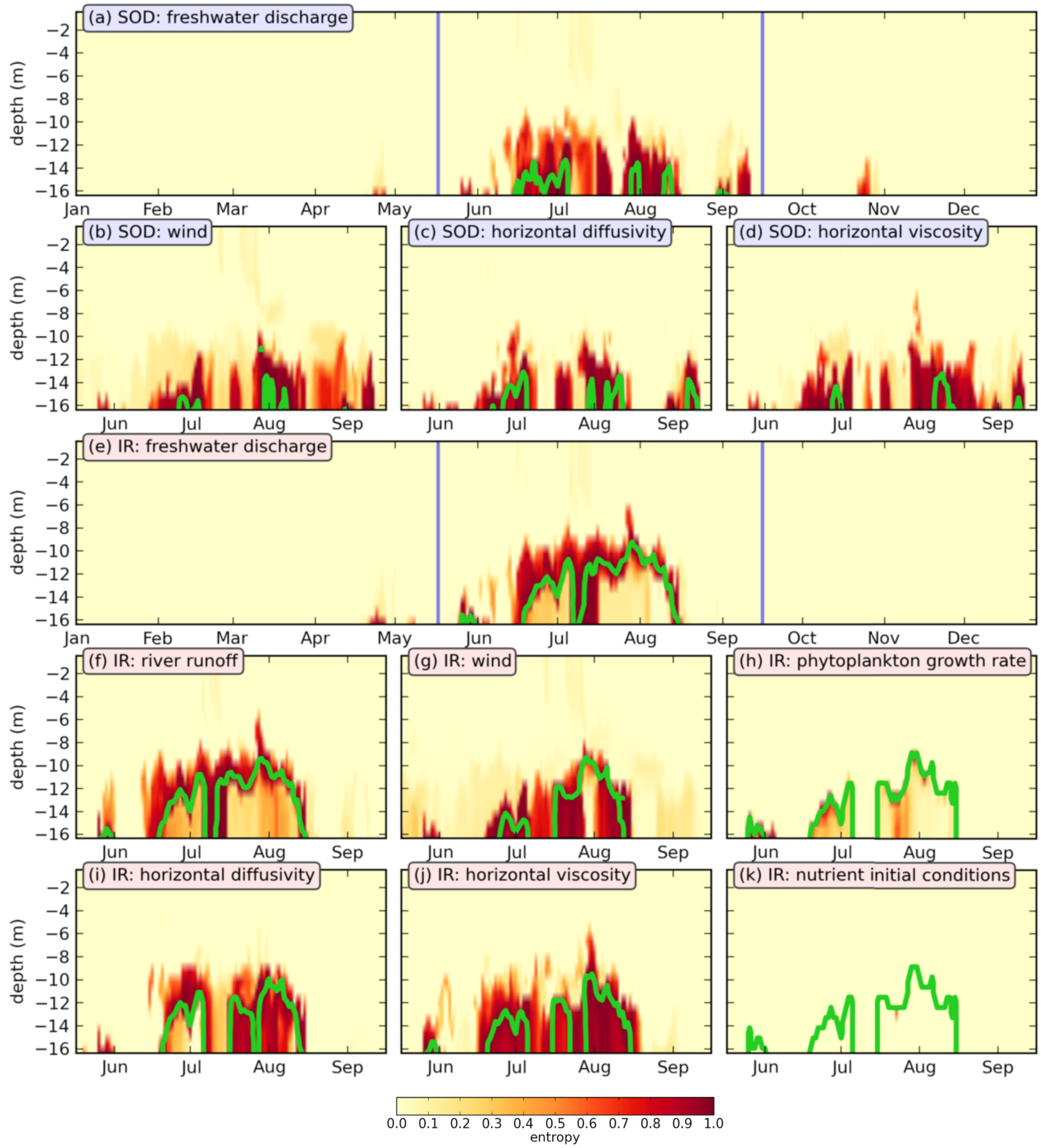


Figure 5.7: Time-depth plots of hypoxia entropy for selected inputs to the SOD and IR configurations (SOD: (a)-(d), IR: (e)-(k)) for the station marked in Figure 5.5. The large panels ((a) and (e)) show the development of entropy for the entire year 2004, smaller panels focus on the time span mid-May to mid-September (marked by the blue, vertical lines in (a) and (e)). Regions outlined in green have a probability > 0.5 of being hypoxic. The inputs that are not shown in this figure typically exhibit low uncertainty in oxygen estimates and entropy values close to zero. As a representative of these inputs, I have included the *nutrient initial conditions* input (k).

that have a strong effect on bottom oxygen, but not for those with low effects such as the nutrient initial conditions (as an example, I have included the nutrient initial conditions for the IR configuration in Figure 5.7(k), in the SOD configuration entropy values are equally low).

The IR model configuration features stronger, more persistent hypoxic events at the station (Figure 5.5(d)) which correspond to larger maximum likelihood regions (Figure 5.7(e)-(k)). With the exception of *horizontal viscosity*, all inputs exhibit low entropy values within the maximum likelihood region. These correspond to points where the model is nearly certain that hypoxia will occur.

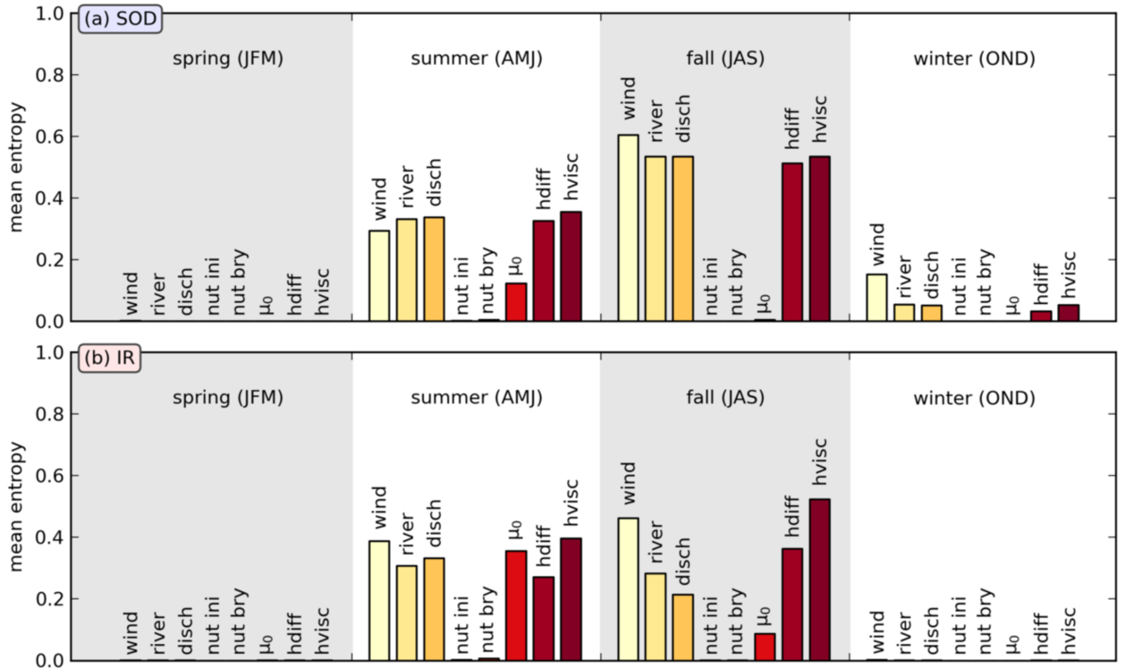


Figure 5.8: Seasonal (time-averaged) hypoxia entropy values for the SOD (a) and IR (b) configurations at the station marked in Figure 5.5.

To summarize the temporal development of entropy, I obtained seasonal averages of the entropy at the station (Figure 5.8). Both model configurations show the same patterns in the first half of 2004: uncertainties due to variations of the input are close to zero early in the year and become significantly larger in April. During the second half of the year, uncertainties are larger in the SOD configuration compared to the IR, with the exception of those caused by the μ_0 input which are generally lower in the SOD configurations due to the tighter coupling between the nitrogen cycle and sediment remineralization that I explained above. Generally, uncertainty values of bottom hypoxia correspond well to the variations caused in the bottom oxygen concentrations (Figure 5.7).

Spatial Distribution of Entropy

The spatial distribution of hypoxia entropy in mid July (Figure 5.9) corresponds well to my previous observations: Regions in the domain away from the coast exhibit low uncertainty and a low probability of hypoxia; high entropy values occur close to the coast and mostly along the fringe of the maximum likelihood hypoxic area. The station (blue star in Figure 5.9) is in this region of high entropy values along the fringe which accounts for the high uncertainty values observed in the temporal development previously (Figure 5.7).

Consistent with the properties of the region of hypoxia in the baseline simulations, the SOD model configuration shows a less homogeneous and more fractured region of high entropy compared to IR (compare Figure 5.9 SOD (a)-(d) to IR (e)-(i)). All model inputs create a region of low entropy within the maximum likelihood hypoxic area close to the Mississippi and Atchafalaya River mouths. These correspond to a high probability of hypoxia and low uncertainty.

Based on the spatial maps of the probability of hypoxia I compute brackets for the size of the hypoxic area on 17 July 2004 (Figure 5.10). For this purpose, I sum the areas of the model bottom grid cells whose probability of hypoxia is greater than a given threshold. For a threshold of 0.5, I obtain the maximum likelihood area of hypoxia. Other thresholds can define uncertainty estimates for the size of the hypoxic area: the interval in between thresholds of 0.6 and 0.4 correspond to the size estimate of the area with entropy values ≥ 0.97 (see Figure 5.3(a)). The wind input, both river inputs and the horizontal mixing coefficients induce wide ranges into the size estimates, especially in the SOD model configuration. For example, for both river inputs, the area between the 0.6 and 0.4 probability threshold is greater than $5\,000$ km^2 which corresponds to more than 40% of the maximum likelihood estimate (Figure 5.10(a)). In the SOD configuration, the sizes of the maximum likelihood areas for the high uncertainty inputs also deviate strongly from the estimates of the baseline simulation without uncertainty. Here it is most apparent that the symmetric distribution of the uncertain input can lead to a highly skewed response in the model.

Overall, the hypoxia entropy estimates correspond well to my previous observations: some inputs have result in large uncertainty in hypoxia, while two inputs (nutrient initial

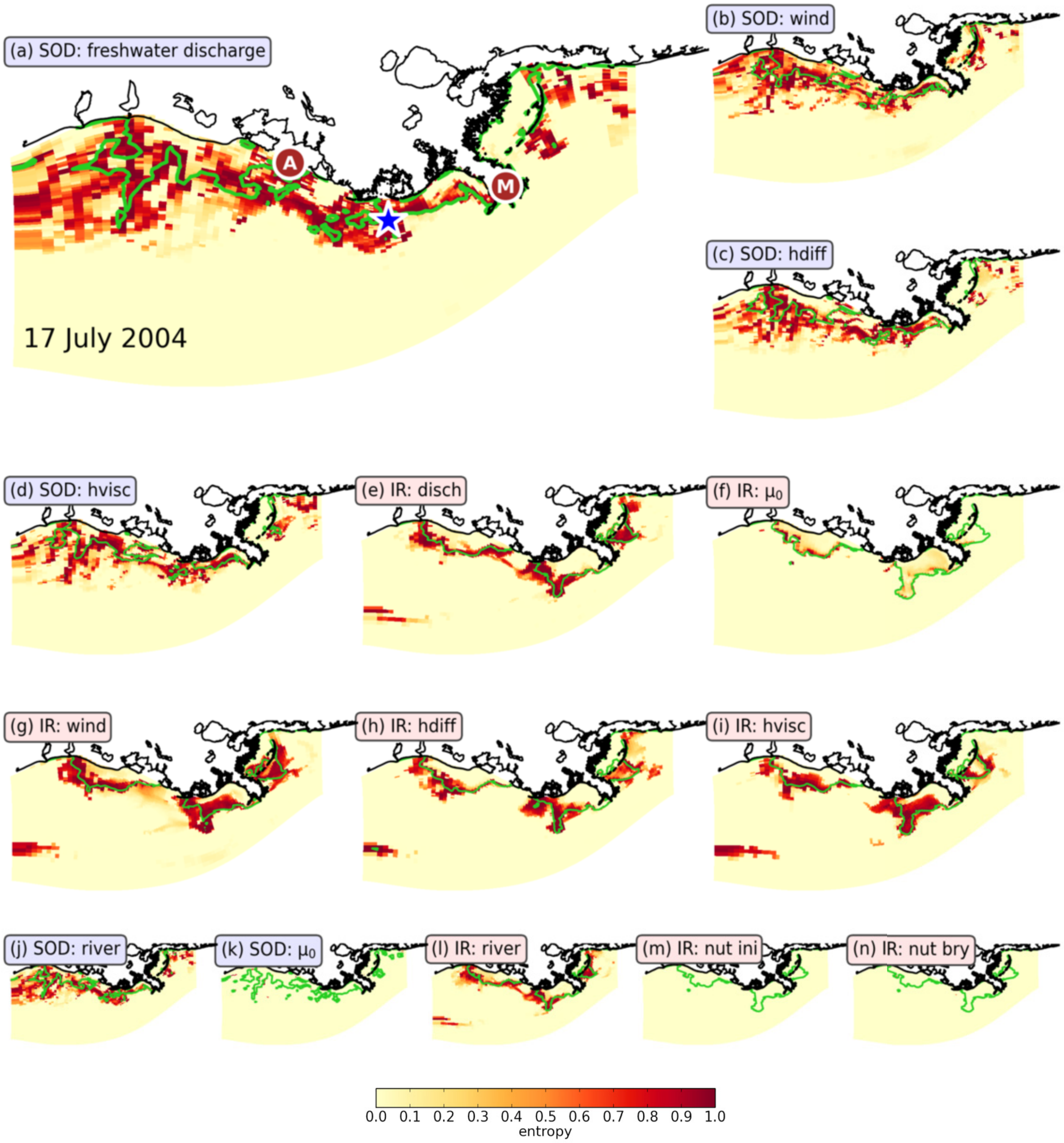


Figure 5.9: Spatial maps of bottom hypoxia uncertainty (entropy) for selected inputs to the SOD and IR configurations on 17 July 2004. Regions outlined in green have a probability > 0.5 of being hypoxic. The bottom row of panels displays inputs with low entropy values and the SOD and IR *river* input maps ((j) and (l)) which are nearly identical to their respective *disch* input maps (a) and (e). The blue star in (a) marks the station corresponding to the time-depth plots in Figure 5.7, the Mississippi and Atchafalaya River mouths are marked by brown circles with “M” and “A”, respectively.

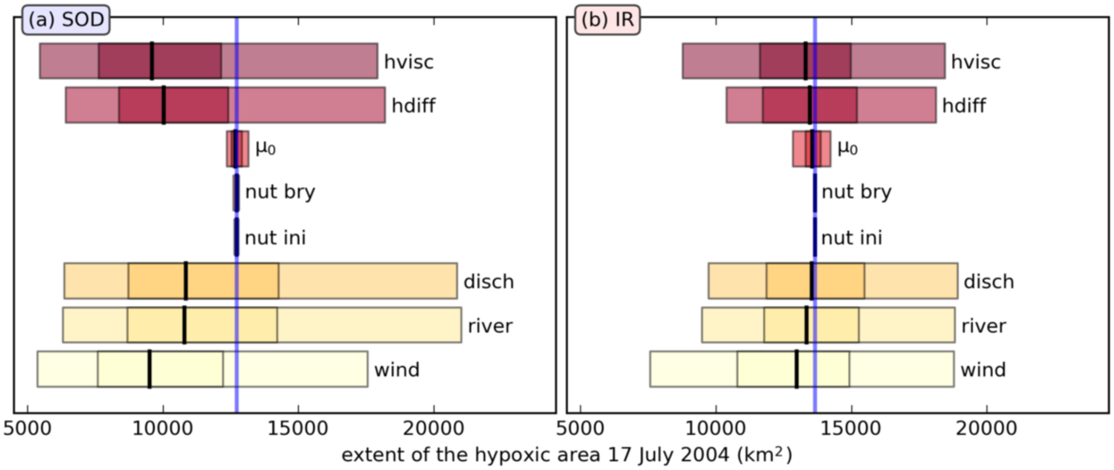


Figure 5.10: The distribution of the size of the hypoxic area on 17 July 2004 induced by uncertainty in the inputs. Each bar corresponds to one uncertain input to the SOD (a) and IR (b) model configuration. The outer bar is the size of the area that has a probability of hypoxia between at least 0.75 (lower limit of outer bar) and at least 0.25 (upper limit of outer bar). The inner bar marks the $[0.4, 0.6]$ probability interval and the black line corresponds to the area with a probability that is at least 0.5, the maximum likelihood area. The blue vertical lines mark the size of the hypoxic area in the SOD and IR baseline simulations without uncertainty. The entropy values corresponding to these probabilities are marked in Figure 5.3. The time development of these properties for the SOD *disch* input are shown in Fig 5.11.

and boundary conditions) have nearly no effect on bottom oxygen. Uncertainty in hypoxia is temporally and spatially constrained, as variation of my uncertain model inputs cannot induce hypoxic conditions early or late in the year, or far offshore from the Mississippi and Atchafalaya River mouths.

5.3.4 Comparison of Hypoxic Area Estimates with Observations

For the freshwater river input I obtained several years of emulator-based uncertainty estimates to derive a time-series of the distribution of the hypoxic area size (Figure 5.11). The series exhibits an annual cycle, the recurrent hypoxic area typically disappears from November to March and reaches its maximum extent in late July or August. The uncertainty in the hypoxic area displays a seasonal cycle, as well, with high uncertainty during the summer months and small uncertainty at times of small hypoxic area estimates. The deviation of the median size of the hypoxic area from the baseline simulation which I observed in Figure 5.10 persists throughout the four years of simulation.

Using the time-series, I attempt a qualitative comparison with hypoxic area estimates which I computed based on bottom oxygen observations from the LUMCON cruises (*Rabalais et al.*, 2007) for the years 2004-2007 (black bars in Figure 5.11). Only in the years 2004 and 2006, the maximum likelihood estimate and the baseline simulation estimate agree with the observation-based estimate (Figure 5.11(a) and (c)). In 2005, variations in the river input cannot explain the deviation between model and observation-based estimates which lies far outside the region with a hypoxia probability of 0.25 or higher (Figure 5.11(b)). In 2007, the data lies within this region. In other words, by extending the maximum likelihood hypoxia area to the area that has a probability of hypoxia of 0.25 or higher (based on the uncertainty in the freshwater river input) I reach an area with a size that corresponds to the observation-based estimate (Figure 5.11(d)). Uncertainties in the freshwater river input can help explain the model misfit here.

It should be noted that the observational estimates contain an error as well due to non-synopticity in the observations, sampling error in the spatially and temporally varying field, and the error associated with interpolation. Nevertheless, simple analyzes like this can help to gain an understanding of model error sources and their effect on model output.

5.3.5 Uncertainty in Surface Chlorophyll Estimates

To explore the effect of uncertain inputs on other model output, I briefly describe the distribution of surface chlorophyll in the model. Surface chlorophyll is often used in data

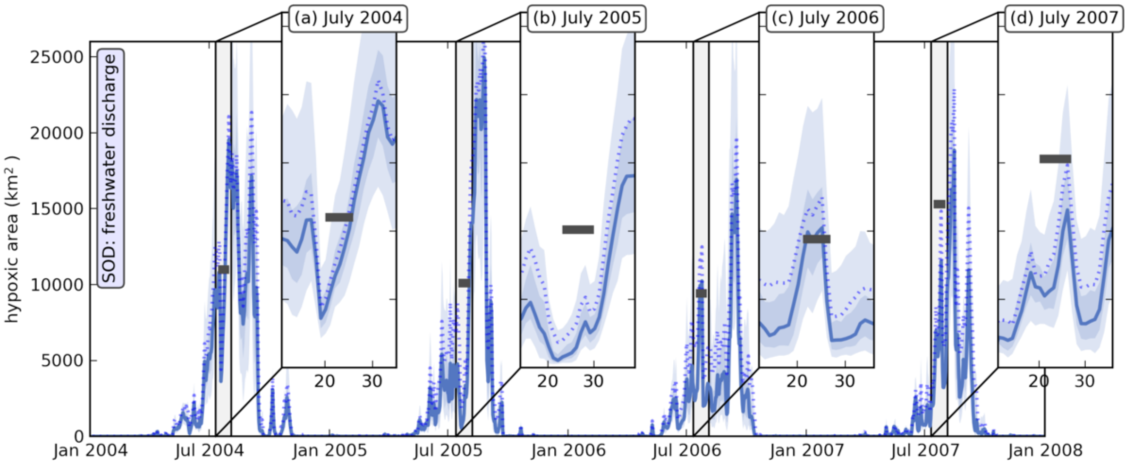


Figure 5.11: Temporal development of the distribution of the size of the hypoxic area for uncertain *disch* input for the SOD model configuration in comparison with estimates based on observations. The coloured regions in the plot correspond to the freshwater river input to the SOD model configuration. The outer region is the size of the area that has a probability of hypoxia between at least 0.75 (lower limit of outer region) and at least 0.25 (upper limit of outer region). The inner corresponds to probability values of 0.6 and 0.4. The thick blue line corresponds to the area with a probability that is at least 0.5, the maximum likelihood size of the hypoxic area. The dotted line shows the development of the size of the hypoxic area in the baseline simulation without uncertainty. The large black bars correspond to estimates of the hypoxic area based on observations from the LUMCON cruises which are performed annually in late July; the width of each bar marks the date range of the corresponding cruise.

assimilation and model validation applications for biological ocean models because of the wealth of available satellite observations (see e.g. *Allen et al.*, 2007; *Mattern et al.*, 2010b). To obtain an overview of the effect of the inputs on surface chlorophyll, I compute the time-averaged seasonal quantiles at the station (Figure 5.12). By comparing the surface chlorophyll quantiles to the bottom oxygen quantiles, it is evident that the same inputs have a strong effect on both quantities, albeit in a different order. Both river inputs have a very pronounced impact on surface chlorophyll at the station and the extra nutrient input of the *river* compared to the *disch* input manifests itself in a slightly more spread out distribution. As with bottom oxygen, seasonal changes in the effect of the inputs are apparent and all of the inputs have a higher impact during summer.

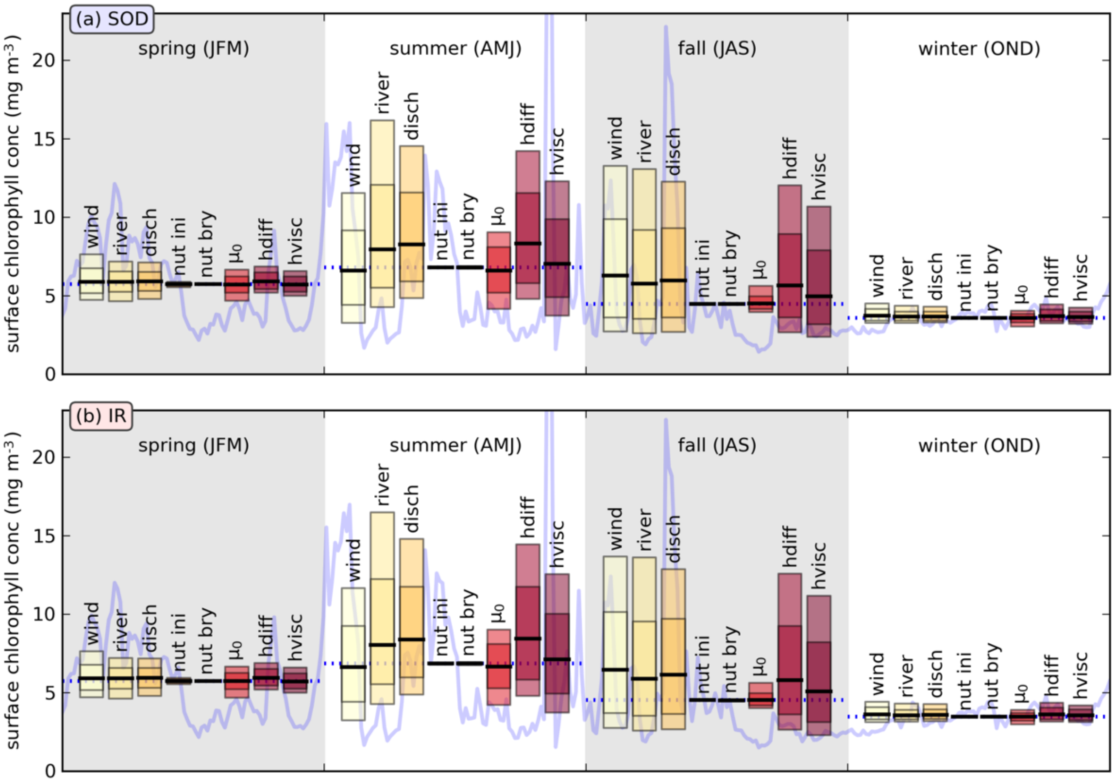


Figure 5.12: Seasonal (time-averaged) quantiles of the surface chlorophyll distribution at the station marked in Figure 5.5 for 2004. Each bar marks the region between the 0.1 and 0.9 quantiles (outer bar), the region between the 0.25 and 0.75 quantiles (inner bar), and the median (solid black line) for one uncertain input, in one season in 2004. The blue solid line in the background marks the development of surface chlorophyll in the baseline simulation without uncertainty; its seasonal averages are displayed as dotted blue lines. In contrast to bottom oxygen (Figure 5.6), differences between the SOD and IR configuration are not very pronounced.

The choice of SOD or IR model configuration has no noticeable impact on the surface chlorophyll field at the station. The effects of the uncertain inputs leads to negligible differences in the time-averaged quantiles (compare Figure 5.12(a) and (b)). This is true even for μ_0, which, unlike its effect on bottom oxygen, has the same effect on surface chlorophyll in the SOD and IR configuration.

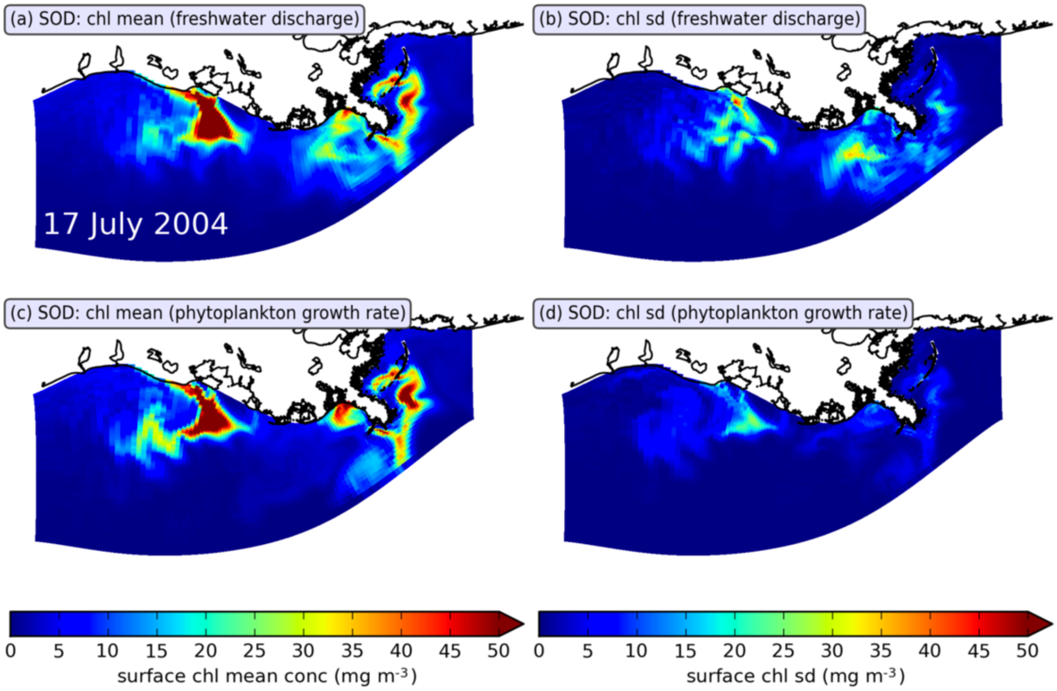

(a) SOD: chl mean (freshwater discharge)
17 July 2004
(b) SOD: chl sd (freshwater discharge)
(c) SOD: chl mean (phytoplankton growth rate)
(d) SOD: chl sd (phytoplankton growth rate)
0 5 10 15 20 25 30 35 40 45 50
surface chl mean conc (mg m-3)
0 5 10 15 20 25 30 35 40 45 50
surface chl sd (mg m-3)

Figure 5.13: Mean and standard deviation (sd) of surface chlorophyll due to uncertainty the *disch* and μ_0 input for the SOD configuration.

To get an idea of the spatial spread of uncertainty, I obtain the mean and standard deviation of the surface chlorophyll field in mid July for the *disch* and μ_0 inputs (Figure 5.13). Both inputs show similar patterns, mean surface chlorophyll is highest close to the Mississippi delta and the mouth of the Atchafalaya where the nutrient input into the model is high. Uncertainty, in form of the standard deviation, is lower for the μ_0 input. For the *disch* input, high values of the standard deviation are found along the fringes of the high chlorophyll areas, similar to the spread of entropy along the boundary of the hypoxic area (Figure 5.9). Variations of the *disch* input appear to affect the surface patch of high chlorophyll in a similar way it affects the hypoxic area at the bottom.

5.4 Discussion

For this chapter, I performed a series of experiments to assess how uncertainty in selected inputs of a physical-biological coastal ocean model propagates into its output. For this purpose I used an emulator approach, the polynomial chaos expansion, to approximate the input and output distributions. I selected 8 uncertain inputs from both the physical and biological model and used the same relative amount of uncertainty for all inputs. On the output side, I focus on uncertainty introduced to the biological model, especially estimates of oxygen and bottom hypoxia. The goals of this chapter were to examine the effect of different inputs in time and space and compare their impacts. Two benthic parametrizations, IR and SOD, were considered in the examination.

The results of this chapter offer insights into the sensitivity of my model and allow for an assessment of the impact of the uncertain inputs at each point in time and space of my model simulations. The most obvious result of the comparison of the different inputs is that two inputs, the nutrient boundary and initial conditions, have very little influence on the bottom oxygen concentration and the surface chlorophyll content in my model, while the model is considerably more sensitive to the other inputs I investigated. Uncertainty in the inputs to the physical model (wind, freshwater river input, vertical mixing coefficients) had the strongest effect on both bottom oxygen and surface chlorophyll. Input to the biological model had generally a much smaller impact, including variations in the river nutrient supply, which in addition to the freshwater impact had little extra effect on the observed model output.

Beside model sensitivity, this chapter provides some insights into model uncertainty, specifically model uncertainty due to uncertainties, or error in the inputs. Here, my analysis cannot provide a full assessment of model uncertainty, since the model response to the inputs were observed individually and not all inputs were considered. Most importantly, I deliberately chose the same relative error for all inputs in order to better compare their effect, instead of selecting a specific distribution for each input. Based on my experiments, I can still draw some important conclusions with regard to model error from the results. Uncertainty in the inputs typically has little effect on oxygen or chlorophyll outside the biologically very productive zone close to the river mouths or outside the spring and summer months (April through September). Within the region and period of higher error, however, uncertainties due to just a single input can be considerably higher. Estimates of hypoxia, for example, are strongly influenced by all of the physical inputs I investigated (see Figure 5.9) and with it size estimates of the hypoxic area (see Figure 5.10). Even if the input uncertainties are decreased (in this chapter, I use distribution with an standard deviation of 20%), it is very likely that their combined impact on bottom oxygen estimates should lead to high errors in hypoxia estimates in regions around the river mouths.

In conjunction with observations, an uncertainty analysis can be expanded in order to find a relationship between model uncertainty and misfit of model and observations. In Section 5.3.4 I contrasted hypoxic area estimates based on observations compared to model output with uncertainty due to river inflow. I have shown that the variations I introduced into the river inflow can in three out of four cases explain the discrepancy with the observations. Assuming that the error in the observations is relatively small, the uncertainty cannot explain the discrepancy that occurs in one year. Here, an adjustment of the river inflow in the model would thus likely not lead to much improved fit to the observations and one has to turn to other sources of model uncertainty for improvement. I have included this comparison with observations to illustrate just one of many ways to integrate observations into an uncertainty analysis (see e.g. *Allen et al.*, 2007).

As I have shown in Section 5.2.6.1, the emulator-based uncertainty estimates themselves are not free of error. While uncertainties in the biological inputs are propagated into the output reliably, there is a higher error when estimating the model response to variations in the physical inputs (Figure 5.4). Lower emulation error can be achieved only by adding more quadrature points, i.e. increasing the number of samples from the input distribution, yet this comes at a much higher computational cost. What is interesting in regard to the emulator error is that it spatially correlates with the region of high model uncertainty (compare Figure 5.4(b) and Figure 5.9). The cause of the error is that the emulator does not sample the output on a sufficiently dense grid to capture its variability. One reason for this model behaviour is that, in contrast to the biological inputs, changes in the physical inputs can affect the stochastic flow field, e.g. instabilities and eddies in the model, leading to highly variable small scale changes. It is this high model sensitivity to the physical inputs that leads to the emulator error. For this reason, I hypothesize that the high emulator error does not cause a systematic overestimation of the model error, but rather that both model and emulator errors are caused by the same phenomenon, the highly variable model output. Thus the emulator estimate is right at least qualitatively by assigning high uncertainty values to the regions in question.

In order to quantify the uncertainty in model estimates of hypoxia I used entropy, a simple function of the probability of hypoxia (Figure 5.3(a)). In cases where I am interested in a binary events (the oxygen at a grid cell concentration can either be below the hypoxic threshold or not) entropy is the suitable way to quantify uncertainty (*Shannon*, 2001). High entropy values that I encounter at many occasions (see Fig 5.7 or 5.9) indicate that the state of the grid cell, hypoxic or not hypoxic, is highly uncertain. As conversion errors from probability to entropy are reduced at high entropy values (Figure 5.3(b)), I have higher confidence in these high uncertainty values. This is further evidence that model-based prediction of hypoxia is difficult in specific model regions.

5.5 Conclusions

I performed an extensive analysis of the impact of 8 physical and biological inputs on a physical-biological model of the Texas-Louisiana shelf in the northern Gulf of Mexico. The goal of the analysis was to assess model sensitivity and uncertainty with a focus on the biological model output, in particular bottom oxygen and hypoxia. For the purpose of propagating uncertainty from model input to its output, I used a model emulator, the polynomial chaos expansion which offered a straightforward approach to propagate uncertainty. While the computational cost of an emulator-based uncertainty analysis is significant, it offers the possibility of a thorough assessment of model sensitivity for a select number of model inputs.

I found that the impact of the different inputs varied considerably: the physical inputs typically had a stronger effect than the biological inputs, with variations in the freshwater river input and the wind field having the strongest impact on both bottom oxygen but also surface chlorophyll. Two of the biological inputs, nutrient initial and boundary conditions, showed negligible effect on the output. Strong effect of all outputs is constrained to the biologically active regions in both time and space. The uncertainty induced into the system is considerable and especially affects estimates of hypoxia and the hypoxic area. For example, hypoxic area estimates of the model vary by more than $5\,000$ km^2 or by around 40% (probability of hypoxia between 0.4 and 0.6) due to 20% variation in the river inflow.

CHAPTER 6

CONCLUSIONS

In this thesis several methods for the assimilation of observations into biological ocean models are presented and assessed. Chapter 2 lays the groundwork for data assimilation by examining eight image comparison measures, adapted for use with satellite images of surface chlorophyll. The AGB distance, which showed the best performance throughout various tests was used for data assimilation in the following Chapters 3 and 4. Chapter 3 introduces the use of an emulator for observation-based state and parameter estimation in a biological ocean model. In Chapter 4 the same model and observations are used to assess the capabilities of particle filter-based data assimilation for both state and parameter estimates. Finally, the emulator from Chapter 3 provides the means to estimate model sensitivity and uncertainty based on prescribed uncertainty in the model inputs. In this chapter, various outstanding issues in data assimilation are discussed that were motivated by the work in this thesis.

Due to the complexity of realistic, biological ocean models, the greatest impediment to application of statistical procedures is the computational cost of the model simulations. Nonlinear and non-Gaussian statistical procedures, such as the particle filter in Chapter 4, rely on sample-based solutions, i.e. they require samples to approximate properties of a particular distribution. The number of samples determines the quality of the statistical approximation, yet each sample requires an expensive model simulation, severely limiting the number of samples that can be obtained. One could argue that, with increases in computational power, the cost of model runs will decrease in the future, allowing more and faster model evaluations. It can be expected, however, that model complexity will grow along with the increases in computational capabilities. Thus computational constraints, requiring small ensemble and sample sizes, will likely remain important factors influencing the effectiveness of statistical procedures.

Computational constraints played an important role throughout this thesis and affected many implementation choices in the previous chapters:

1. Image comparison based on the AGB distance introduced in Chapter 2 is the basis for data assimilation in this work. From a computational perspective, the AGB distance can be evaluated very quickly, relative to the simulation time of the ocean models. Even though many AGB evaluations were required for this work, these have negligible impact on the overall computational cost of the state and parameter estimation techniques presented here.
2. The emulator approach used in Chapters 3 and 5 allows for the approximation of model output at comparatively low computational cost, permitting efficient computation of state, parameter and uncertainty estimates. Yet, the emulator itself is based on model output and an initial set of model simulations is required for the emulator to operate on. This required initial set of model runs limits the possible field of applications of the emulator. For example, eight inputs are analyzed in the emulator-based sensitivity analysis in Chapter 5; their individual analysis took $7 \times 8 = 56$ model simulations, while a joint analysis would push computational cost to $7^8 = 5\,764\,801$ model simulations and beyond feasibility. Thus, while the emulator can increase computational efficiency considerably, it cannot eliminate computational constraints. Generally, the use of an emulator is highly beneficial in cases where many model evaluations are required for just a few variable inputs; it becomes less useful as fewer model evaluations are needed or more variable inputs are considered.
3. As a consequence of the high computational complexity for particle filters and other ensemble-based data assimilation procedures ensembles are limited to small sizes. Small ensembles lead to undersampling of the high dimensional model state probability distribution. This is especially evident in the presence of outliers (see Section 4.3.4) and becomes problematic in high-dimensional applications of particle filters where it is one important cause of ensemble collapse (*van Leeuwen*, 2009). In the particle filter implementation in Chapter 4, the use of an error subspace reduces the number of dimensions the particle filter operates in strongly (from potentially more than 2 million to two). The error subspace approach implicitly relies on the assumption that model error is due to just a few sources. In the application in Chapter 4, it is assumed that error in modelled chlorophyll resides in two sources, both of them parameters of the biological model. This approach is limited,

because there are many more sources of model error, and the variation that is introduced to the two parameters cannot account for all deviations between model and observations (see Figure 4.4). Despite these drawbacks, the error subspace approach is practical, can be extended to more error sources and allows for the successful application of the particle filter to high-dimensional models and realistic observations using small ensemble sizes. This approach is especially suitable for biological ocean models which often contain several unknown parameters that have high uncertainty.

In practice, particle filters and other data assimilation procedures that are partially based on Bayesian principles are less commonly used than alternative procedures such as the EnKF and variational techniques. The latter two use Gaussian and linear approximations to the Bayesian solution and are very commonly found in oceanographic data assimilation. Direct comparisons show that particle filter techniques can achieve better results in nonlinear applications, such as typical ocean models, but the EnKF is more efficient computationally (*Annan and Hargreaves*, 2010). An approach which increases the efficiency of the particle filter is the introduction of the error subspace, as discussed above. In addition, the use of ADA in the particle filter implementation reduces the number of independent observations and thereby the probability of ensemble collapse (*Snyder et al.*, 2008; *van Leeuwen*, 2009). Other recent approaches aim to add other efficiency-increasing modifications to the basic particle filter algorithm (*van Leeuwen*, 2010, 2011) or combine characteristics of Bayesian techniques with those based on Gaussian approximations (*Bocquet et al.*, 2010). More work is required to test these techniques in realistic applications and in the face of observation error which can have a serious effect on data assimilation (see Section 4.4.3).

The particle filter introduced in Chapter 4 is a technique designed for data assimilation. Originally formulated for state estimation, it can easily be extended to estimate parameters as well. In contrast, the emulator is essentially a surrogate for the model, a quick way to approximate model output without having to perform new simulations. Only by comparing the emulator output with observations, e.g. via minimizing the AGB distance between emulator results and observations (see Section 3.2.4), can state and parameter estimates be obtained. In Chapters 2 and 4, emulator and particle filter are used in exactly the same scenario: using the same model and set of observations estimates for the same state variables and parameters are obtained. The direct comparison shows that, both state and parameter estimates from the emulator approach are superior to that of the particle filter (compare e.g. chlorophyll estimates in Figure 3.7 to Figures 4.3 and 4.4, or parameter estimates in Figure 3.7). Here, the emulator's way of sampling parameters in a non-random fashion appears to be better suited for state estimation and especially parameter estimation compared to the Monte Carlo-based random sampling and resampling performed by the particle filter. In addition, the emulator allows for new state and parameter estimates, for example, based on a localized subset of the observations (see Section 3.3.4.1), without the need for new model simulations. In comparison, a new set of observations would require a full new particle filter experiment. However, the improved performance of the emulator in this scenario comes at a computational price: while the particle filter can obtain its results with 10 ensemble members, the results of the emulator are based on 49 model runs. Additionally, the particle filter can potentially examine more parameters and other model inputs at a relatively low increase in ensemble size, compared to the emulator which exhibits exponential growth in model simulations with respect to the number of inputs. The particle filter can also adjust the ensemble probability distribution if an unsuitable initial distribution is specified (see Section 4.4.4). The emulator, in contrast, is bound to its initial parameter ranges, and new model simulations would be required if the initial simulations cannot represent the model state distribution adequately.

The methods introduced in Chapters 2 through 5 are very versatile and their application is not restricted to specific models or types of observations. The image comparison techniques can, for example, be applied in other scenarios involving the comparison of spatial fields which may contain features at different scales, noise and missing values. The data assimilation procedures presented in this work also pose few restrictions on the model they can be applied to: neither particle filter nor emulator involve changes to the code of the model and neither technique requires the model to be linear or the model error to have a specific distribution. Thus emulator and particle filter are very versatile and allow for a simple implementation with different models. While the same set of chlorophyll satellite observations and the same image comparison method are used in this work, these can be replaced by other types of data in conjunction with suitable model-data distance measures without fundamental changes to the algorithms presented here.

Parameter estimation is of special interest for data assimilation in biological ocean models. Both, emulator approach and particle filter allow for efficient parameter estimation in complex models. One key advantage of both techniques is that the required model simulations can be performed in parallel. For complex models which take a long time to evaluate, this can lead to a substantial decrease in runtime compared to parameter estimation techniques that require sequential model evaluations, for example variational data assimilation techniques using gradient descent-based optimization. While the number of parameters that can be estimated is limited by the computational constraints discussed above, considerable improvement can be achieved by optimizing just a few key parameters of biological ocean models (see Chapter 3).

The application of the emulator and particle filter techniques in this work offers some insights into typical biological ocean models. The most striking finding is the temporal development of the optimal values of the two plankton parameters that were examined in Chapters 3 and 4. By assimilating satellite images of chlorophyll at a daily frequency, both the emulator-based approach and the particle filter found a time-dependence in form of a seasonal cycle in the optimal parameter values. Moreover, letting the model parameters follow this seasonal cycle leads to a substantial improvement in the model prediction of chlorophyll. As discussed in Section 3.4, this finding points to seasonal processes that are not adequately represented in the model with fixed parameters. A future step would be to use a longer set of satellite observations to test if this seasonal development of the biological parameters repeats over multiple years. If so, the model with time-dependent parameters could be used for forecasting and could be considered a medium-complexity alternative to highly complex biological models with many more phytoplankton variables, advocated for example by *Follows and Dutkiewicz* (2011).

The sensitivity and uncertainty analysis in Chapter 5 highlights the variability of biological model output based on uncertainty in the biological and physical model inputs. It is evident that model properties at the ocean surface and the sea floor are subject to considerable amounts of uncertainty based on various inputs. Especially the physical inputs cause strong variability in the output, which affects both small scale features, such as the oxygen content in a specific grid cell (see Figure 5.6), as well as large scale properties, such as the size of the hypoxic area (see Figure 5.10). Unlike the biological inputs, the physical inputs can cause perturbations in the stochastic flow field, increasing uncertainty in the model output and decreasing the quality of the emulator approximation (see Figure 5.4). These results indicate high model uncertainty of key variables of biological models, it also highlights the strong dependence of biological properties on the underlying physical model.

In summary, this work presents the successful application of state, parameter and uncertainty estimation techniques to complex biological ocean models. Challenges and limitations of the techniques are discussed and solutions offered, all in relation to realistic data assimilation scenarios with realistic models and real observations. Besides testing the methodology, this work also offers insights into the models to which the methods are applied.

APPENDIX A

COPYRIGHT

1. A modified version of **Chapter 2** was published in the *MDPI* journal *Remote Sensing* and is reproduced here with permission:

> All articles published by MDPI are made available under an open access license worldwide immediately. This means: everyone has free and unlimited access to the full-text of all articles published in MDPI journals, and everyone is free to re-use the published material given proper accreditation/citation of the original publication.

(See `http://www.mdpi.com/about/openaccess`.)

2. An edited version of **Chapter 3** was published by *Elsevier* in the *Journal of Marine Systems*. It is reproduced here with permission:

> What rights do I retain as a journal author? [...] the right to include the journal article, in full or in part, in a thesis or dissertation; [...]

(See `http://www.elsevier.com/wps/find/authorsview.authors/rights`.)

BIBLIOGRAPHY

Alberga, V., Similarity Measures of Remotely Sensed Multi-Sensor Images for Change Detection Applications, *Remote Sensing*, *1*, 122–143, 2009.

Allen, J. I., M. Eknes, and G. Evensen, An Ensemble Kalman Filter with a complex marine ecosystem model: hindcasting phytoplankton in the Cretan Sea, *Annales Geophysicae*, *21*, 399–411, 2003.

Allen, J. I., J. T. Holt, J. Blackford, and R. Proctor, Error quantification of a high-resolution coupled hydrodynamic-ecosystem coastal-ocean model: Part 2. Chlorophyll-a, nutrients and SPM, *Journal of Marine Systems*, *68*, 381–404, 2007.

Annan, J., Modelling under uncertainty: Monte Carlo methods for temporally varying parameters, *Ecological Modelling*, *136*, 297–302, 2001.

Annan, J., and J. Hargreaves, Efficient identification of ocean thermodynamics in a physical/biogeochemical ocean model with an iterative Importance Sampling method, *Ocean Modelling*, *32*, 205–215, 2010.

Askey, R., and J. A. Wilson, *Some basic hypergeometric orthogonal polynomials that generalize Jacobi polynomials*, vol. 319, Memoirs of the American Mathematical Society, 1985.

Aulenbach, B., H. Buxton, W. Battaglin, and R. Coupe, Streamflow and nutrient fluxes of the Mississippi-Atchafalaya River basin and subbasins for the period of record through 2005, *Tech. rep.*, U. S. Geological Survey, 2007.

Aumont, O., E. Maier-Reimer, S. Blain, and P. Monfray, An ecosystem model of the global ocean including Fe, Si, P colimitations, *Global Biogeochemical Cycles*, *17*, 1060, 2003.

Avcıbaş, İ., B. Sankur, and K. Sayood, Statistical evaluation of image quality measures, *Journal of Electronic Imaging*, *11*, 206–223, 2002.

Barlow, R. G., R. F. C. Mantoura, M. A. Gough, and T. W. Fileman, Pigment signatures of the phytoplankton composition in the northeastern Atlantic during the 1990 spring bloom, *Deep Sea Research Part II: Topical Studies in Oceanography*, *40*, 459–477, 1993.

Béal, D., P. Brasseur, J.-M. Brankart, Y. Ourmières, and J. Verron, Characterization of mixing errors in a coupled physical biogeochemical model of the North Atlantic: implications for nonlinear estimation using Gaussian anamorphosis, *Ocean Science*, *6*, 247–262, 2010.

Beezley, J. D., and J. Mandel, Morphing ensemble Kalman filters, *Tellus A*, *60*, 131–140, 2008.

Belkin, I. M., and J. E. O'Reilly, An algorithm for oceanic front detection in chlorophyll and SST satellite imagery, *Journal of Marine Systems*, *78*, 319–326, 2009.

Bennett, A., *Inverse modeling of the ocean and atmosphere*, Cambridge University Press, 2002.

Bianchi, T., S. DiMarco, J. C. Jr., R. Hetland, P. Chapman, J. Day, and M. Allison, The science of hypoxia in the Northern Gulf of Mexico: A review, *Science of The Total Environment*, *408*, 1471–1484, 2010.

Bianucci, L., K. L. Denman, and D. Ianson, Low oxygen and high inorganic carbon on the Vancouver Island Shelf, *Journal of Geophysical Research*, *116*, C07011, 2011.

Blatman, G., and B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, *Probabilistic Engineering Mechanics*, *25*, 183–197, 2010.

Bocquet, M., C. A. Pires, and L. Wu, Beyond Gaussian Statistical Modeling in Geophysical Data Assimilation, *Monthly Weather Review*, *138*, 2997–3023, 2010.

Box, G., and N. Draper, *Empirical model-building and response surfaces.*, John Wiley & Sons, 1987.

Cayula, J.-F., and P. Cornillon, Edge Detection Algorithm for SST Images, *Journal of Atmospheric and Oceanic Technology*, *9*, 67–80, 1992.

Chan, A., Comparative Physiological Study Of Marine Diatoms And Dinoflagellates In Relation To Irradiance And Cell-size. II. Relationship Between Photosynthesis, Growth, And Carbon-chlorophyll A-ratio, *Journal of Phycology*, *16*, 428–432, 1980.

Chen, K., and R. He, Numerical Investigation of the Middle Atlantic Bight Shelfbreak Frontal Circulation Using a High-Resolution Ocean Hindcast Model, *Journal of Physical Oceanography*, *40*, 949–964, 2010.

Chorin, A. J., and X. Tu, Implicit sampling for particle filters, *Proceedings of the National Academy of Sciences*, *106*, 17249–17254, 2009.

Ciavatta, S., R. Torres, S. Saux-Picart, and J. I. Allen, Can ocean color assimilation improve biogeochemical hindcasts in shelf seas?, *Journal of Geophysical Research*, *116*, C12043, 2011.

Clancy, D., J. E. Tanner, S. McWilliam, and M. Spencer, Quantifying parameter uncertainty in a coral reef model using Metropolis-Coupled Markov Chain Monte Carlo, *Ecological Modelling*, *221*, 1337–1347, 2010.

Denman, K. L., Modelling planktonic ecosystems: parameterizing complexity, *Progress In Oceanography*, *57*, 429–452, 2003.

Di Gesú, V., and S. Roy, Pictorial Indexes and Soft Image Distances, in *Advances in Soft Computing AFSS 2002*, edited by N. Pal and M. Sugeno, vol. 2275 of *Lecture Notes in Computer Science*, pp. 63–79, Springer Berlin / Heidelberg, 2002.

Di Gesú, V., and V. Starovoitov, Distance-based functions for image comparison, *Pattern Recognition Letters*, *20*, 207–214, 1999.

Doney, S., D. Glover, and R. Najjar, A new coupled, one-dimensional biological-physical model for the upper ocean: Applications to the JGOFS Bermuda Atlantic time-series study (BATS) site, *Deep-Sea Research Part II - Topical Studies in Oceanography*, *43*, 591–624, 1996.

Doucet, A., S. Godsill, and C. Andrieu, On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing*, *10*, 197–208, 2000.

Dowd, M., A sequential Monte Carlo approach for marine ecological prediction, *Environmetrics*, *17*, 435–455, 2006.

Dowd, M., Bayesian statistical data assimilation for ecosystem models using Markov Chain Monte Carlo, *Journal of Marine Systems*, *68*, 439–456, 2007.

Dowd, M., Estimating parameters for a stochastic dynamic marine ecological system, *Environmetrics*, *22*, 501–515, 2011.

Dubuisson, M., and A. Jain, A modified Hausdorff distance for object matching, in *Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing*, vol. 1, pp. 566–568, 1994.

Evensen, G., The Ensemble Kalman Filter: theoretical formulation and practical implementation, *Ocean Dynamics*, *53*, 343–367, 2003.

Evensen, G., *Data assimilation: the ensemble Kalman filter*, Springer Verlag, 2009.

Evensen, G., and P. J. van Leeuwen, An Ensemble Kalman Smoother for Nonlinear Dynamics, *Monthly Weather Review*, *128*, 1852–1867, 2000.

Feng, Y., S. F. DiMarco, and G. A. Jackson, Relative role of wind forcing and riverine nutrient input on the extent of hypoxia in the northern Gulf of Mexico, *Geophysical Research Letters*, *39*, L09601, 2012.

Fennel, K., and J. Wilkin, Quantifying biological carbon export for the northwest North Atlantic continental shelves, *Geophysical Research Letters*, *36*, L18605, 2009.

Fennel, K., J. Wilkin, J. Levin, J. Moisan, J. O'Reilly, and D. Haidvogel, Nitrogen cycling in the Middle Atlantic Bight: Results from a three-dimensional model and implications for the North Atlantic nitrogen budget, *Global Biogeochemical Cycles*, *20*, GB3007, 2006.

Fennel, K., J. Wilkin, M. Previdi, and R. Najjar, Denitrification effects on air-sea CO(2) flux in the coastal ocean: Simulations for the northwest North Atlantic, *Geophysical Research Letters*, *35*, L24608, 2008.

Fennel, K., R. Hetland, Y. Feng, and S. DiMarco, A coupled physical-biological model of the Northern Gulf of Mexico shelf: model description, validation and analysis of phytoplankton variability, *Biogeosciences*, *8*, 1881–1899, 2011.

Field, C. B., M. J. Behrenfeld, J. T. Randerson, and P. Falkowski, Primary Production of the Biosphere: Integrating Terrestrial and Oceanic Components, *Science*, *281*, 237–240, 1998.

Flagg, C., C. Wirick, and S. Smith, The interaction of phytoplankton, zooplankton and currents from 15 months of continuous data in the mid-atlantic bight, *Deep Sea Research Part II: Topical Studies in Oceanography*, *41*, 411, 1994.

Follows, M. J., and S. Dutkiewicz, Modeling Diverse Communities of Marine Microbes, *Annual Review of Marine Science*, *3*, 427–451, 2011.

Follows, M. J., S. Dutkiewicz, S. Grant, and S. W. Chisholm, Emergent Biogeography of Microbial Communities in a Model Ocean, *Science*, *315*, 1843–1846, 2007.

Franks, P., and C. Chen, Plankton production in tidal fronts: A model of Georges Bank in summer, *Journal of Marine Research*, *54*, 631–651, 1996.

Frolov, S., A. M. Baptista, T. K. Leen, Z. Lu, and R. van der Merwe, Fast data assimilation using a nonlinear Kalman filter and a model surrogate: An application to the Columbia River estuary, *Dynamics of Atmospheres and Oceans*, *48*, 16–45, 2009.

Geider, R., H. MacIntyre, and T. Kana, Dynamic model of phytoplankton growth and acclimation: responses of the balanced growth rate and the chlorophyll a:carbon ratio to light, nutrient-limitation and temperature, *Marine Ecology Progress Series*, *148*, 187–200, 1997.

Geider, R., H. MacIntyre, and T. Kana, A dynamic regulatory model of phytoplanktonic acclimation to light, nutrients, and temperature, *Limnology and Oceanography*, *43*, 679–694, 1998.

Gelman, A., and D. Rubin, Practical Markov Chain Monte Carlo: Rejoinder: Replication without Contrition, *Statistical Science*, *7*, 503–511, 1992.

Gneiting, T., and A. E. Raftery, Weather Forecasting with Ensemble Methods, *Science*, *310*, 248–249, 2005.

Godsill, S. J., A. Doucet, and M. West, Monte Carlo Smoothing for Nonlinear Time Series, *Journal of the American Statistical Association*, *99*, 156–168, 2004.

Goebel, N. L., C. A. Edwards, J. P. Zehr, and M. J. Follows, An emergent community ecosystem model applied to the California Current System, *Journal of Marine Systems*, *83*, 221–241, 2010.

Gordon, N., D. Salmond, and A. Smith, Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *Radar and Signal Processing, IEE Proceedings F*, *140*, 107–113, 1993.

Gregg, W. W., P. Ginoux, P. S. Schopf, and N. W. Casey, Phytoplankton and iron: validation of a global three-dimensional ocean biogeochemical model, *Deep Sea Research Part II: Topical Studies in Oceanography*, *50*, 3143–3169, 2003.

Haidvogel, D., et al., Ocean forecasting in terrain-following coordinates: Formulation and skill assessment of the Regional Ocean Modeling System, *Journal of Computational Physics*, *227*, 3595–3624, 2008.

Holyer, R., and S. Peckinpaugh, Edge detection applied to satellite imagery of the oceans, *Geoscience and Remote Sensing, IEEE Transactions on*, *27*, 46–56, 1989.

Hooten, M., W. Leeds, J. Fiechter, and C. Wikle, Assessing first-order emulator inference for physical parameters in nonlinear mechanistic models, *Journal of Agricultural, Biological, and Environmental Statistics*, *16*, 475–494, 2011.

Hu, J., K. Fennel, J. P. Mattern, and J. Wilkin, Data assimilation with a local Ensemble Kalman Filter applied to a three-dimensional biological model of the Middle Atlantic Bight, *Journal of Marine Systems*, *94*, 145–156, 2012.

Huttenlocher, D., G. Klanderman, and W. Rucklidge, Comparing images using the Hausdorff distance, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *15*, 850–863, 1993.

Jaynes, E. T., Information Theory and Statistical Mechanics, *Physical Review*, *106*, 620–630, 1957.

Jones, E., J. Parslow, and L. Murray, A Bayesian approach to state and parameter estimation in a Phytoplankton-Zooplankton model, *Australian Meteorological and Oceanographic Journal*, *59*, 7–15, 2010.

Juffs, P., E. Beggs, and F. Deravi, A multiresolution distance measure for images, *Signal Processing Letters, IEEE*, *5*, 138–140, 1998.

Kane, J., The demography of Calanus finmarchicus (Copepoda: Calanoida) in the Middle Atlantic Bight, USA, 19772001, *Journal of Plankton Research*, *27*, 401–414, 2005.

Karniadakis, G. E., and J. Glimm, Uncertainty quantification in simulation science, *Journal of Computational Physics*, *217*, 1–4, 2006.

Kim, S., R. M. Samelson, and C. Snyder, Toward an Uncertainty Budget for a Coastal Ocean Model, *Monthly Weather Review*, *139*, 866–884, 2010.

Kitagawa, G., Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, *Journal of Computational and Graphical Statistics*, *5*, 1–25, 1996.

Kitagawa, G., A Self-Organizing State-Space Model, *Journal of the American Statistical Association*, *93*, 1203–1215, 1998.

Lawson, L. M., E. E. Hofmann, and Y. H. Spitz, Time series sampling and data assimilation in a simple marine ecosystem model, *Deep Sea Research Part II: Topical Studies in Oceanography*, *43*, 625–651, 1996.

Le Moigne, J., and J. Tilton, Refining image segmentation by integration of edge and region data, *Geoscience and Remote Sensing, IEEE Transactions on*, *33*, 605–615, 1995.

Lehmann, M. K., K. Fennel, and R. He, Statistical validation of a 3-D bio-physical model of the western North Atlantic, *Biogeosciences*, *6*, 1961–1974, 2009.

Lehmann, T., A. Sovakar, W. Schmiti, and R. Repges, A comparison of similarity measures for digital subtraction radiography, *Computers in Biology and Medicine*, *27*, 151–167, 1997.

Lermusiaux, P. F., Uncertainty estimation and prediction for interdisciplinary ocean dynamics, *Journal of Computational Physics*, *217*, 176–199, 2006.

Losa, S. N., G. A. Kivman, J. Schröter, and M. Wenzel, Sequential weak constraint parameter estimation in an ecosystem model, *Journal of Marine Systems*, *43*, 31–49, 2003.

Lucas, D. D., and R. G. Prinn, Parametric sensitivity and uncertainty analysis of dimethylsulfide oxidation in the clear-sky remote marine boundary layer, *Atmospheric Chemistry and Physics*, *5*, 1505–1525, 2005.

Makler-Pick, V., G. Gal, M. Gorfine, M. R. Hipsey, and Y. Carmel, Sensitivity analysis for complex ecological models A new approach, *Environmental Modelling &amp; Software*, *26*, 124–134, 2011.

Mannino, A., M. Russ, and S. Hooker, Algorithm development and validation for satellite-derived distributions of DOC and CDOM in the US Middle Atlantic Bight, *Journal of Geophysical Research*, *113*, C07051, 2008.

Marra, J., R. Houghton, and C. Garside, Phytoplankton Growth At The Shelf-break Front In The Middle Atlantic Bight, *Journal of Marine Research*, *48*, 851–868, 1990.

Marzouk, Y. M., and H. N. Najm, Dimensionality reduction and polynomial chaos acceleration of Bayesian inference in inverse problems, *Journal of Computational Physics*, *228*, 1862–1902, 2009.

Mattern, J. P., M. Dowd, and K. Fennel, Sequential data assimilation applied to a physical-biological model for the Bermuda Atlantic time series station, *Journal of Marine Systems*, *79*, 144–156, 2010a.

Mattern, J. P., K. Fennel, and M. Dowd, Introduction and Assessment of Measures for Quantitative Model-Data Comparison Using Satellite Images, *Remote Sensing*, *2*, 794–818, 2010b.

Mattern, J. P., K. Fennel, and M. Dowd, Estimating time-dependent parameters for a biological ocean model using an emulator approach, *Journal of Marine Systems*, *9697*, 32–47, 2012.

Matre, O. L., O. Knio, H. Najm, and R. Ghanem, Uncertainty propagation using Wiener-Haar expansions, *Journal of Computational Physics*, *197*, 28–57, 2004.

McKay, M. D., R. J. Beckman, and W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, *Technometrics*, *21*, 239–245, 1979.

Melbourne-Thomas, J., C. Johnson, and E. Fulton, Characterizing sensitivity and uncertainty in a multiscale model of a complex coral reef system, *Ecological Modelling*, *222*, 3320–3334, 2011.

Mitchell, B. G., Coastal zone color scanner retrospective, *Journal of Geophysical Research*, *99*, 7291–7292, 1994.

Moore, J. K., S. C. Doney, J. A. Kleypas, D. M. Glover, and I. Y. Fung, An intermediate complexity marine ecosystem model for the global domain, *Deep Sea Research Part II: Topical Studies in Oceanography*, *49*, 403–462, 2001.

Nichol, D., Autonomous Extraction of an Eddy-Like Structure from Infrared Images of the Ocean, *Geoscience and Remote Sensing, IEEE Transactions on*, *GE-25*, 28–34, 1987.

Palmer, T., G. Shutts, R. Hagedorn, F. Doblas-Reyes, T. Jung, and M. Leutbecher, Representing model uncertainty in weather and climate prediction, *Annu. Rev. Earth Planet. Sci.*, *33*, 163–193, 2005.

Powell, B. S., H. G. Arango, A. M. Moore, E. Di Lorenzo, R. F. Milliff, and D. Foley, 4DVAR data assimilation in the intra-Americas sea with the Regional Ocean Modeling System (ROMS), *Ocean Modelling*, *23*, 130–145, 2008.

Previdi, M., K. Fennel, J. Wilkin, and D. Haidvogel, Interannual variability in atmospheric CO2 uptake on the northeast U.S. continental shelf, *Journal of Geophysical Research*, *114*, G04003, 2009.

Rabalais, N., R. Turner, B. Sen Gupta, D. Boesch, P. Chapman, and M. Murrell, Characterization and long-term trends of hypoxia in the northern Gulf of Mexico: does the science support the Action Plan, *Estuaries and Coasts*, *30*, 753–772, 2007.

Riley, G., Factors controlling phytoplankton populations on Georges Bank, *Journal of Marine Research*, *6*, 54–73, 1946.

Ristic, B., S. Arulampalam, and N. Gordon, *Beyond the Kalman Filter: Particle Filters for Tracking Applications*, Artech House, 2004.

Rougier, J., and D. M. Sexton, Inference in ensemble experiments, *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *365*, 2133–2143, 2007.

Sakov, P., G. Evensen, and L. Bertino, Asynchronous data assimilation with the EnKF, *Tellus Series A-Dynamic Meteorology and Oceanography*, *62*, 24–29, 2010.

Santini, S., and R. Jain, Similarity measures, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *21*, 871–883, 1999.

Scott, V., H. Kettle, and C. J. Merchant, Sensitivity analysis of an ocean carbon cycle model in the North Atlantic: an investigation of parameters affecting the air-sea CO_2 flux, primary production and export of detritus, *Ocean Science*, *7*, 405–419, 2011.

Shannon, C. E., A mathematical theory of communication, *ACM SIGMOBILE Mobile Computing and Communications Review*, *5*, 3–55, 2001.

Shen, C. Y., T. E. Evans, and S. Finette, Polynomial Chaos Quantification of the Growth of Uncertainty Investigated with a Lorenz Model, *Journal of Atmospheric and Oceanic Technology*, *27*, 1059–1071, 2010.

Sim, D.-G., O.-K. Kwon, and R.-H. Park, Object matching algorithms using robust Hausdorff distance measures, *IEEE Transactions on Image Processing*, *8*, 425–429, 1999.

Smedstad, O., and J. J. O'Brien, Variational data assimilation and parameter estimation in an equatorial Pacific Ocean model, *Progress in Oceanography*, *26*, 179–241, 1991.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson, Obstacles to High-Dimensional Particle Filtering, *Monthly Weather Review*, *136*, 4629–4640, 2008.

Stow, C. A., J. Jolliff, D. J. M. Jr., S. C. Doney, J. I. Allen, M. A. Friedrichs, K. A. Rose, and P. Wallhead, Skill assessment for coupled biological/physical models of marine systems, *Journal of Marine Systems*, *76*, 4–15, 2009.

Thacker, W., A. Srinivasan, M. Iskandarani, O. Knio, and M. Le Hnaff, Propagating boundary uncertainties using polynomial expansions, *Ocean Modelling*, *4344*, 52–63, 2012.

van Leeuwen, P. J., A Variance-Minimizing Filter for Large-Scale Applications, *Monthly Weather Review*, *131*, 2071–2084, 2003.

van Leeuwen, P. J., Particle Filtering in Geophysical Systems, *Monthly Weather Review*, *137*, 4089–4114, 2009.

van Leeuwen, P. J., Nonlinear data assimilation in geosciences: an extremely efficient particle filter, *Quarterly Journal of the Royal Meteorological Society*, *136*, 1991–1999, 2010.

van Leeuwen, P. J., Efficient nonlinear data-assimilation in geophysical fluid dynamics, *Computers & Fluids*, *46*, 52–58, 2011.

Wan, X., and G. Karniadakis, Beyond Wiener-Askey expansions: Handling arbitrary PDFs, *Journal of Scientific Computing*, *27*, 455–464, 2006.

Wang, L., Y. Zhang, and J. Feng, On the Euclidean distance of images, *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *27*, 1334–1339, 2005.

Wiener, N., The Homogeneous Chaos, *American Journal of Mathematics*, *60*, 897–936, 1938.

Wikle, C. K., and L. M. Berliner, A Bayesian tutorial for data assimilation, *Physica D: Nonlinear Phenomena*, *230*, 1–16, 2007.

Wilson, D. L., A. J. Baddeley, and R. A. Owens, A New Metric for Grey-Scale Image Comparison, *International Journal of Computer Vision*, *24*, 5–17, 1997.

Xiu, D., and G. Karniadakis, The Wiener-Askey polynomial chaos for stochastic differential equations, *SIAM Journal on Scientific Computing*, *24*, 619–644, 2002.

Xiu, D., and G. E. Karniadakis, Modeling uncertainty in flow simulations via generalized polynomial chaos, *Journal of Computational Physics*, *187*, 137–167, 2003.