Florian Beiser

Local Drift Forecasting with Simplified Ocean Models and Multi-Level Data Assimilation

Thesis for the degree of Philosophiae Doctor

Trondheim, March 2024

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

NTNU | Kunnskap for en bedre verden

NTNU
Norwegian University of Science and Technology

Thesis for the degree of Philosophiae Doctor

Faculty of Information Technology and Electrical Engineering
Department of Mathematical Sciences

ISBN XXX-XX-XXX-XXXX-X (printed version)
ISBN XXX-XX-XXX-XXXX-X (electronic version)
ISSN 1503-8181

Doctoral theses at NTNU, XXXX:XX

Abstract

Search-and-rescue operations at sea are supported by fast predictions of drift trajectories that are classically based on operational ocean models. This thesis promotes a complementary approach for drift forecasting based on computationally efficient methodologies utilising simplified models and ensemble-based data assimilation.

Simplified ocean models aim to capture only the most relevant dynamics for short time horizons and they are hence more computationally efficient than complex operational models. Herein, the rotational shallow-water equations and a massively parallel simulation framework are exploited for the simplified modelling. Given the inherent inaccessibility of the true dynamics of the ocean, both presently and in the future, large ensembles of simplified models can be run to account for this spatio-temporal uncertainty in local forecasts. Such ensemble-based representations enable the incorporation of observations of ocean currents by data assimilation techniques as new measurements are available. Consequently, the uncertainty in the prediction is typically reduced. In this work, methodologies behind such an on-demand system for local short-term drift trajectory prediction are considered. The investigations include different modelling and assimilation techniques suitable for search-and-rescue scenarios.

The first part of this thesis synthesises the background and the description of the general concepts, whereas the second part consists of the scientific papers. The contributions in this thesis reach from the discussion of numerical solvers for shallow-water simulations, over mathematical modelling for simplified ocean dynamics, to tailored data assimilation methods for sparse in-situ observations and settle with the advancement of computational efficient data assimilation, building on the foundations of multilevel Monte Carlo methods and simulations on different resolutions.

Contents

Preface

This thesis is submitted in partial fulfillment of the requirements for the degree of philosophiae doctor (PhD) at the Norwegian University of Science and Technology (NTNU), Trondheim, Norway.

The work has been performed at the Department of Mathematics and Cybernetics, SINTEF Digital, Oslo, and the Department of Mathematical Sciences, NTNU, Trondheim. It was financed by the Norwegian Research Council through the Havvarsel project, grant number 310515.

Acknowledgments

Først og fremst vil jeg framheve Håvards veiledning som førte til denne avhandlingen. Jeg er veldig takknemlig for at han sprer så mye motivasjon, lagånd og gode råd. Jeg setter stor pris på det tette samarbeidet preget av spontane diskusjoner med lav terskel og de mange tipsene relatert til vitenskapelig arbeid, programmering og for å holde blikk på de store målene i prosjektet. Dessuten takker jeg Jo for å ha ønsket meg velkommen på NTNU i Trondheim i det første semesteret og for å ha holdt kontakten gjennom hele doktorgradstudiet. Det var nyttig med innspill fra den statistiske siden fra Jo og vurderingen fra den operative siden fra Martin i MET. I tillegg var det artig å jobbe sammen med Kjetil og å diskutere med ham matematikken som ligger bak. Ved siden av det faglige har Knut-Andreas bistod med å skape de riktige rammebetingelsene, slik at jeg har kunnet lære mange ting i løpet av de siste tre og et halvt årene.

For øvrig har kollegene i Applied Computational Science-gruppen i SINTEF Digital støttet alltid med et trivelig og svært profesjonelt miljø, og ikke minst med mange ideer for nye eventyr i den norske naturen. Likeså var det gøy å tilbringe tid med gode kollegaer på IMF hos NTNU. Takk til alle sammen!

Florian Beiser
Oslo, February 2024

List of Papers

Papers included in thesis

I: **Reducing Numerical Artifacts by Sacrificing Well-Balance for Rotating Shallow-Water Flow**
Håvard Heitlo Holm, Florian Beiser
Finite Volumes for Complex Applications X—Volume 2, Hyperbolic and Related Problems

DOI: 10.1007/978-3-031-40860-1_19

II: **Combining Barotropic and Baroclinic Simplified Models for Drift Trajectory Predictions**
Florian Beiser, Håvard Heitlo Holm, Martin Lilleeng Sætra, Nils Melsom Kristensen, Kai Håkon Christensen
In review

III: **Comparison of Ensemble-Based Data Assimilation Methods for Sparse Oceanographic Data**
Florian Beiser, Håvard Heitlo Holm, Jo Eidsvik
Quarterly Journal of the Royal Meteorological Society

DOI: 10.1002/qj.4637

IV: **Multi-Level Data Assimilation for Simplified Ocean Models**
Florian Beiser, Håvard Heitlo Holm, Kjetil Olsen Lye, Jo Eidsvik
In review

Part I

Background

Chapter 1

Introduction

Daily weather forecasts are a familiar service and they extensively influence people's everyday lives on multiple aspects. For everyone at sea or in coastal areas, ocean forecasts are equally important as atmospheric weather forecasts. While numerical weather prediction and ocean forecasting share structural similarities, the preconditions for ocean forecasts are more challenging: the dynamics are more complex due to feature systems of smaller scales and the amount of observation data is significantly less.

Data plays a crucial role to improve the description of the current conditions of the ocean, but observations alone are insufficient to recover the complete ocean state and fall short to predict future ones. Therefore, observations are combined with numerical simulations. Mathematical models resemble the dominating physical processes in the ocean and generate forecasts for future dynamics, while available measurements are sequentially assimilated into model forecasts. After an update reflecting the latest observations, new forecasts can be produced. As more data is assimilated closer to the target time, the precision of statistical predictions is thereby improved.

At sea, search-and-rescue (SAR) operations rely on fast and trustworthy forecasts of drift trajectory to define search areas [27, 99]. Such drift predictions are commonly produced by trajectory models using the operational forecasts. In that, operational ocean forecast architectures are built on complex models that aim for as precise as possible simulations and assimilation of all available data. Therefore, operational forecasts require large computational resources and deliver predictions on a fixed resolution.

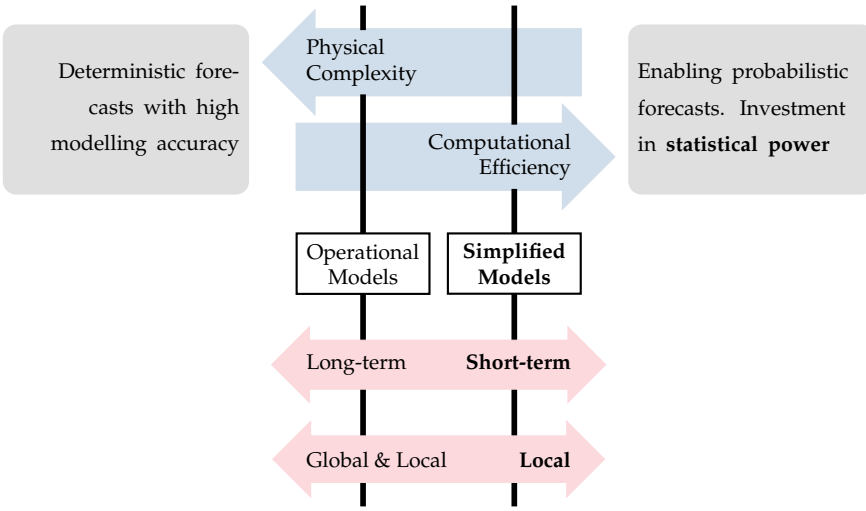


Figure 1.1: Simplified models treat some of the physical complexity in favour for computational efficiency which is useful for conducting uncertainty quantification.

In Norway, 80% of the population lives within 10 km of the coast. Additionally, the Norwegian coast line is the second longest in the world with narrow fjords and archipelagos. For these reasons, high-resolution forecasts that capture local features in the ocean currents and that can be used for rapid drift predictions are required.

In this thesis, we follow a complementary approach for local drift trajectory forecasting that utilises ensembles of simplified models together with multi-level data assimilation. Thereby, we facilitate lightweight decision-support tools that exploit fast and flexible models and assimilate sparse in-situ observations.

In general, ocean circulation models are capable of representing a wide range of oceanographic processes and provide high-fidelity predictions. These cutting-edge ocean models afford an accurate description of the three-dimensional physics, and assimilation of data from different sources for precise estimates of the true ocean state. However, they often produce only a single simulation due to extensive computational demands and restricted resources. Drift forecasts can then be generated by an offline trajectory model using results of operational simulations.

In contrast, simplified ocean models offer a lightweight supplement with reduced computational burden, enabling rapid and replicated simulations. Simplified models capture the dominant physical processes and

Figure 1.2: Photos of experimental drifters used for drift trajectory monitoring during a cruise in Inner Oslofjord on the 27.04.2023. Drifters can be used to gather current observations in a specific area of interest.

can flexibly increase resolutions, making them suitable for local short-term forecasting. They provide a valuable approximation of the real ocean state over short time spans, allowing for efficient and timely predictions in applications like drift trajectory forecasting for SAR or emergency response decision-support. This is a different use case than climate modelling where the goals are involving long time horizons and global coverage [5]. Furthermore, by running numerous simulations, an ensemble of simplified models can feature enhanced statistical properties, efficiently explore the uncertainty space, and yield improved probabilistic forecasts, a capability not easily achievable with deterministic circulation models. Figure 1.1 exhibits these contrasts.

Observational data obtained from, e.g. drifters (see Figure 1.2), can be used to improve the accuracy of ensemble forecasts by data assimilation techniques. Hereby, relevant in-situ observations are sparse, since related measurements are collected only at few locations in the ocean. Data assimilation helps to calibrate models, to correct biases, and adjusts the model state to better match observations, yielding more reliable forecasts which often have reduced uncertainty. With deterministic models on one end of the scale, machine learning models represent another extreme approach where all physical behaviour is supposed to be learned from data [65]. While the machine learning approach often suffers from insufficient training data in the case of SAR missions, data assimilation is a data-driven approach that combines the physical model and observational data in a statistically principled framework.

The computational benefits of simplified ocean models facilitate pro-

ducing ensemble forecasts in local domains with high spatial resolution and assimilating latest in-situ observations that have not been respected in the operational machinery. Consequently, we promote the application of the simplified approach in time-sensitive SAR scenarios.

In this work, we advance the complementary approach that combines simplified ocean models and data assimilation for the prediction of drift trajectories in local areas. Therefore, we present contributions to the following research objectives:

I Adaption of the numerical simulations for simplified ocean models to produce more realistic results.

II Extension of the modelling framework for simplified ocean models to capture a variety of simplified dynamics.

III Dwelling on the data assimilation aspect with techniques that are tailored for sparse observations.

IV Design of multi-level data assimilation for simplified ocean models where simulations on multiple resolutions are exploited.

The *havvarsel* Project The *havvarsel* project is a collaboration among the Norwegian Institute of Marine Research (HI), the Norwegian Meteorological Institute (MET Norway) and SINTEF Digital, and aims to prepare next-generation ocean forecast systems that provide personalised ocean forecasts in coastal zones. This PhD project was carried out within the frame of the *havvarsel* project and builds academically on the work of Holm [72], which was carried out during the previous *GPU Ocean* project. In the preceding project, a GPU-accelerated simplified ocean model that runs for real conditions of the Norwegian coast was implemented and state-of-the-art data assimilation methods for drift trajectories were explored. As technical results of the *havvarsel* project, a two-way data flow system has been established which handles up- and download of individual ocean observations and the topic of ocean forecasting is presented to the public on the web page `https://havvarsel.no`. Complementary, this PhD project focuses on the research and the fundamental development of simplified models and associated data assimilation techniques that are necessary to incorporate such data to improve local forecasts in the future. To demonstrate how decision-support can be advanced, SAR scenarios are considered in this thesis.

Outline The remainder of this thesis is structured as follows. First, Chapter 2 presents the simplified ocean models used throughout this thesis, expounding the shallow-water equations, numerical methods for solving them, and drift trajectory modelling. The Monte Carlo approach to quantify inherent uncertainty by ensembles is explained in Chapter 3 and Chapter 4 introduces ensemble-based data assimilation methods. Here, the data assimilation problem is stated, followed by a discussion of particle filters and ensemble Kalman filters that strive to modify uncertainty realistically in the forecasts. Chapter 5 explains multi-level Monte Carlo estimation along with the multi-level version of the ensemble Kalman filter. Chapter 6 summarises the contributions made in the attached papers and Chapter 7 concludes with a general summary and an outlook.

Chapter 2

Simplified Ocean Models

In the context of geophysical fluid dynamics, mathematical modelling aims at describing the flow processes and relies on the principles of conservation laws, i.e. conservation of mass, momentum, internal energy, and tracer quantities. While the Reynolds-averaged Navier-Stokes equations serve as a starting point when studying ocean dynamics [63], additional approximations are commonly introduced. The hydrostatic approximation assumes that the horizontal scale is large compared to the vertical scale[1], while the Boussinesq approximation assumes the incompressibility of ocean water. These approximations lead to the formulation of the primitive equations for the ocean dynamics. For most operational forecasts, the primitive equations are solved in one way or another by fully three-dimensional circulation models such as ROMS [123], NEMO [97], or HYCOM [11].

These models find also application in the operational forecasting systems of MET Norway. For instance, the TOPAZ system encompasses the North Atlantic region with a resolution of 12–16 km [119], while the NorKyst model is nested into the TOPAZ domain to cover specifically the Norwegian coast, offering a higher resolution of 800 m [2]. NorKyst is the main operational forecast tool for the Norwegian coastal area and is set up with boundary and forcing information from several sources and with 35 vertical layers. In addition, NorFjord forecasts, that are nested into NorKyst, can provide a 160 m resolution, but are not available operationally [1]. Moreover, specialised products can integrate additional services like a sea ice model for Arctic waters [115]. Given the complexity of these models and their computational requirements, supercomputer in-

[1] In the perspective of the hydrostatic approximation, the ocean is shallow while a glass of water is deep.

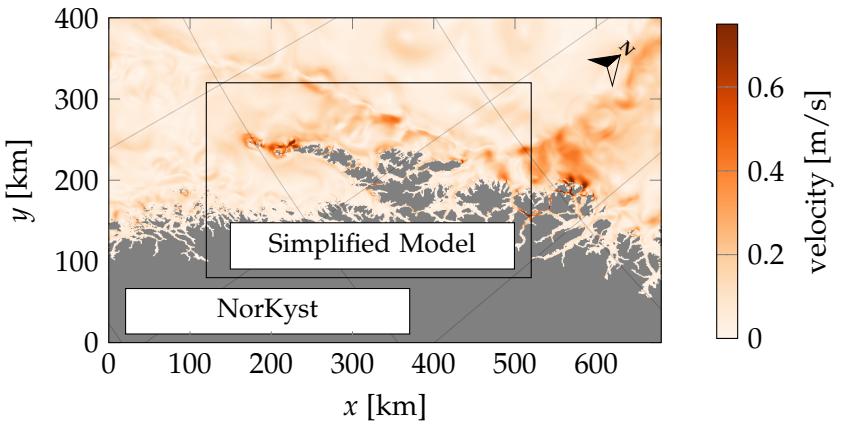


Figure 2.1: NorKyst is the ROMS-based model for the coast of Norway. Simplified models can be nested into its domain locally. Here, we show a cut-out for the Lofoten area in Northern Norway.

frastructure has to be employed, and model runs are typically scheduled only once or twice a day.

These operational forecasts are the backbone of the ocean forecast services [132]. However, when it comes to capturing information of high spatial resolution on-demand, one has to compete for additional computational resources beyond the allocated slots for the scheduled runs. This can take time and these efforts are often limited to a single high-detail deterministic forecast.

Simplified models provide a complementary approach: While they do not attempt to produce the most detailed dynamics, these models are only valid for short-term physics and can be run efficiently. The resolution can be increased for improved representation of local conditions and the models can run on-demand as no large clusters are needed. Simplified models for the Norwegian coast can be nested locally into NorKyst, see Figure 2.1. Then, the simplified model inherits features from NorKyst through the initialisation or from the boundary. Simplified models are less complex and potentially only two-dimensional, such that they can run on conventional computers, for example on board of a vessel.

2.1 Shallow-Water Equations

The primitive equations of ocean dynamics can be further simplified by assuming constant density and by integrating the variables over the full

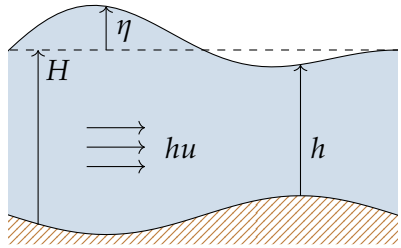


Figure 2.2: Cross section explaining the variables of the shallow-water equations. The momentum hu is the current u integrated over the entire water column $h = H + \eta$, where H is an equilibrium depth and η is the deviation from this equilibrium.

water column, see e.g. [114]. This results in the formulation of the *shallow-water equations (SWE)* within a rotating frame of reference, which constitutes our simplified ocean model in all our further studies. The rotational SWE takes into account the Coriolis force that arises due to the Earth's rotation.

We consider the equilibrium water depth H and the deviation from this equilibrium η, such that the total water column height becomes $h = H + \eta$. The velocities in x- and y-direction are denoted as u and v, respectively (see Figure 2.2). Then the SWE can be expressed as

$$\begin{bmatrix} \eta \\ hu \\ hv \end{bmatrix}_t + \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}_x + \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ fhv \\ -fhu \end{bmatrix} + \begin{bmatrix} 0 \\ ghH_x \\ ghH_y \end{bmatrix}, \quad (2.1)$$

where g is the gravitational force and f the Coriolis parameter. These equations encapsulate the two-dimensional dynamics governed by the evolution of the sea surface elevation and the momenta (hu and hv), where the source terms on the right-hand side account for the effects of the Coriolis force and the variations in the bottom topography. In practice, further source terms can be added, accounting for atmospheric pressure forcing, bed friction, wind stress, and so on.

A lake-at-rest with flat surface and no flow constitutes a trivial example of a steady-state solution to the SWE. However, in the presence of Coriolis forces, an important class of equilibria is described by *geostrophic balance*, where the Coriolis force balances the pressure gradient. This concept yields non-trivial rotating steady state solutions. As the SWE capture the Coriolis

force as a source term and the pressure forces from the slope of the free surface, they have the conceptual capability to sustain these solutions of the type

$$fhv = gh\frac{\partial \eta}{\partial x} \quad \text{and} \quad fhu = -gh\frac{\partial \eta}{\partial y} \tag{2.2}$$

over time. Geostrophic balance plays a significant role in physical oceanography [134] and will arise in several of the examples throughout the thesis.

High-Resolution Finite-Volume Methods

The SWE are a non-linear hyperbolic conservation law. With the vector of conserved variables $\mathbf{q} = [\eta, hu, hv]^\top$, the SWE can be expressed in vectorised form as

$$\mathbf{q}_t + \boldsymbol{\mathcal{F}}(\mathbf{q})_x + \boldsymbol{\mathcal{G}}(\mathbf{q})_y = \boldsymbol{S}(\mathbf{q}). \tag{2.3}$$

Here, $\boldsymbol{\mathcal{F}}(\mathbf{q})$ and $\boldsymbol{\mathcal{G}}(\mathbf{q})$ represent the flux functions in the x- and y- direction, respectively, and $\boldsymbol{S}(\mathbf{q})$ denotes the source terms. To numerically solve this system, we employ *finite-volume methods* (FVM), which are designed to preserve physical properties of the original equations, i.e. FVMs can be composed to be conservative, see e.g. [133, 92, 102]. In our setting, we discretise the computational domain into a regular Cartesian grid of size $n_x \times n_y$, with each grid cell representing a control volume of dimensions $\Delta x \times \Delta y$. Then, we define $\mathbf{Q}_{j,k}$ as the average of the conserved variables $\mathbf{q}$ over the cell $\Gamma_{j,k}$ with the indices (j, k) in the grid. We omit additional time notation as long as it is not strictly necessary.

With Gauss's theorem, eq. (2.3) can be rephrased into its integral form

$$\frac{\partial \mathbf{Q}_{j,k}}{\partial t} = -\int_{\partial\Gamma_{j,k}} \boldsymbol{\mathcal{F}}(\mathbf{q}) \cdot \mathbf{n}_x \, \mathrm{d}s - \int_{\partial\Gamma_{j,k}} \boldsymbol{\mathcal{G}}(\mathbf{q}) \cdot \mathbf{n}_y \, \mathrm{d}s + \boldsymbol{S}(\mathbf{Q}_{j,k}), \tag{2.4}$$

where the integrals are evaluated over the boundary of the cell and $\mathbf{n}_x$ and $\mathbf{n}_y$ are x- and y-contribution of the outer normal vector of the cell, respectively.

FVMs utilise the integral form of eq. (2.3) and approximate the spatial derivatives by the differences of the fluxes over opposite cell faces. With numerical fluxes $\mathbf{F}$ and $\mathbf{G}$, which approximate the flux terms over the cell faces in x- and y-direction, respectively, the semi-discrete approximation to eq. (2.3) becomes

$$\frac{\partial \mathbf{Q}_{j,k}}{\partial t} = -\frac{\mathbf{F}_{j+1/2,k} - \mathbf{F}_{j-1/2,k}}{\Delta x} - \frac{\mathbf{G}_{j,k+1/2} - \mathbf{G}_{j,k-1/2}}{\Delta y} + \boldsymbol{S}(\mathbf{Q}_{j,k}). \tag{2.5}$$

This is a set of ordinary differential equations that describes the evolution of the cell averages in time and is a fundamental building block for many modern FVMs.

One approach to proceed from eq. (2.5) is to utilise the so-called REA-algorithm, which involves three key steps: reconstruction, evolution and averaging. First, a polynomial is reconstructed per cell which enables a better representation of underlying functions. This provides also point values on the faces for the evaluation of the numerical fluxes. Next, the equations are evolved in time using appropriate time integration schemes. Lastly, the FVM representation is again obtained by averaging the results over the grid cells.

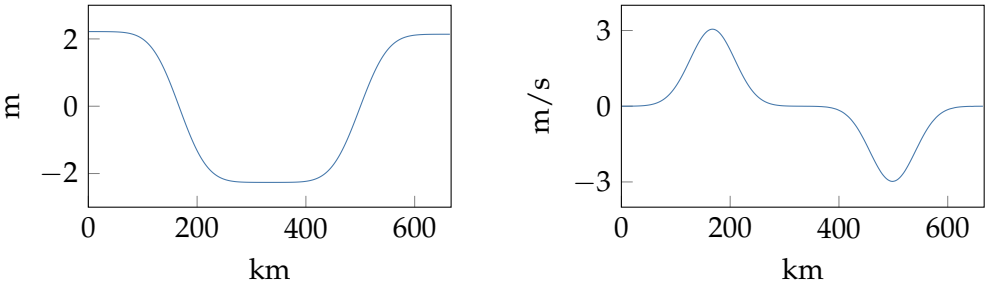
As geostrophic balance is a central concept in oceanography, special effort is exerted in the design of the FVMs to guarantee that also the numerical solutions preserve these steady states. Such FVMs are then called *well-balanced*. Well-balanced schemes are a wider class of FVMs, but we are particularly interested in those that are well-balanced with respect to an instance of geostrophic balance.

In this thesis, we touch on two well-balanced FVMs for the SWE. These are high-resolution schemes of second order that are derived from the REA-algorithm. The first method by Kurganov and Petrova (KP, [87]) is constructed to be well-balanced for the lake-at-rest case, even in the presence of discontinuous bottom topography. KP employs linear reconstruction on the conserved variables and finds slopes using the generalised minmod limiter [95]. The numerical fluxes are then evaluated by the central-upwind scheme [88].

The second scheme by Chertock et al. (CDKLM, [25]) is similar to KP, but it extends the well-balanced property to a larger subset of rotating equilibria. It is proposed for a special case of non-trivial geostrophic jets along the coordinate axes, including the trivial lake-at-rest case. In contrast to KP, the reconstruction is based on potential energies with respect to the geostrophic imbalance and further, to obtain the well-balance for the jets, the components of $\mathbf{F}$ and $\mathbf{G}$ that embrace the huv-term are evaluated with a standard upwind scheme.

For the evolution in time, both methods employ the explicit second-order total-variation-diminishing Runge-Kutta scheme for the time evolution [60], whose time step size Δt is restricted by the Courant-Friedrich-Levy (CFL)-condition, given by

$$\Delta t \leq \frac{C}{4} \min_{j,k} \left\{ \frac{\Delta x}{\max \left| u_{j,k} \pm \sqrt{g h_{j,k}} \right|}, \frac{\Delta y}{\max \left| v_{j,k} \pm \sqrt{g h_{j,k}} \right|} \right\}, \tag{2.6}$$



(a) η **(b)** u

Figure 2.3: Cross section along the y-axis through exemplary initial conditions of the synthetic SWE experiment. Note that $v = 0$.

where $C < 1$ is the Courant number and grid values come from $[\eta_{j,k}, hu_{j,k}, hv_{j,k}]^\top = \mathbf{Q}_{j,k}$.

Finally, to solve eq. (2.1), it is necessary to combine the SWE with appropriate initial and boundary condition. In the context of simplified ocean simulations, the initial conditions are imposed from operational models without spin-up phase at the starting time of the simulation. Analogously, the boundary conditions are set over the time span of the simulation from the operational forecast data, where a flow relaxation scheme accounts for the origin from a mismatching model [33].

Synthetic SWE Experiment Throughout the contributions of this thesis, we consider the so-called double jet case to test numerical schemes and data assimilation methods. Galewsky et al. [51] suggested a test case for numerical solvers for the SWE on a rotating frame of reference. The case starts from an initial steady state in form of a jet, but the system is unstable in the sense that a small perturbation leads to turbulent behaviour. The set-up was extended by Holm et al. [76] by a second jet in the opposite direction and periodic boundary conditions. With that, the case gets more demanding and creates near-realistic currents. The state is described by a westward jet in the north and an eastward jet in the south, both along the coordinate axes. The initial conditions, see Figure 2.3 for an example, are designed to be in geostrophic balance and the example is therefore within the subset for which the CDKLM scheme is well-balanced. Due to it chaotic response to perturbations, the synthetic double jet case serves also as a challenging test case for data assimilation scenarios in the remainder of this thesis.

GPU OCEAN Code Base

While central processing units (CPUs) serve as the primary general-purpose units of a computer system responsible for executing instructions and performing calculations, graphics processing unit (GPUs) can be used to accelerate computations through massive parallelisation. The GPU was originally developed to accelerate and enhance the rendering and displaying of visual content and are therefore designed to handle typical computations on a grid of pixels in a highly parallel manner. As GPUs became programmable, the potential of parallelism could be harvested for general-purpose scientific computing tasks beyond graphics processing [14]. Whereas CPUs typically consist of dozens of computational cores, modern GPUs consist of several thousands of simpler cores operating in the 'single-instruction, multiple-data' paradigm of parallel computing. This means that GPUs excel at problems that can be solved through embarrassingly parallel algorithms, where the same instruction set can be applied to a large set of input data to produce output that are independent from other output values [16].

Due to the explicit time-stepping used to solve eq. (2.1), the FVMs for simulating the SWE are perfect examples of parallel algorithms that are well-suited for GPU accelerations [66, 127, 74]. Over the recent years, the Applied Computational Sciences group (formerly Computational Geosciences group) at SINTEF Digital, together with the Ocean and Ice research unit at MET Norway, have developed the GPU OCEAN code base as tool for simplified ocean forecasts. Therefore, the simulation framework includes among others GPU-accelerated implementations of the KP and CD-KLM schemes for the efficient simulation of the SWE and supports realistic scenarios by allowing initialisation from operational ocean forecasts, such as NorKyst or similar models, and handling domains with land masks. The capabilities include further to fetch boundary conditions and derive forcing terms for wind and atmospheric pressure from operational models. Furthermore, the framework offers the flexibility to adjust the resolution independent of the initial data resolution, accommodates for ensemble simulations and opens for the integration of data assimilation.

Computational Efficiency

The SWE, as simplified two-dimensional ocean model, are designed for specific ocean processes and treat some of the physical complexity in favour for computational efficiency. Still, the SWE can inherit dynamics from the complex models through initial and boundary conditions and preserve

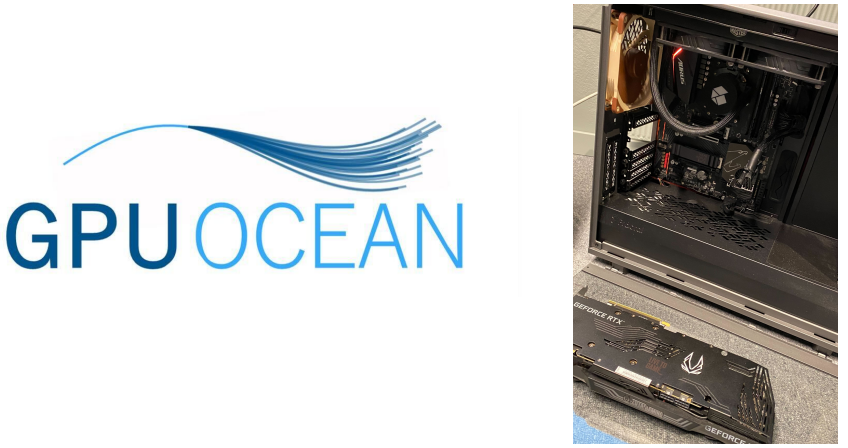
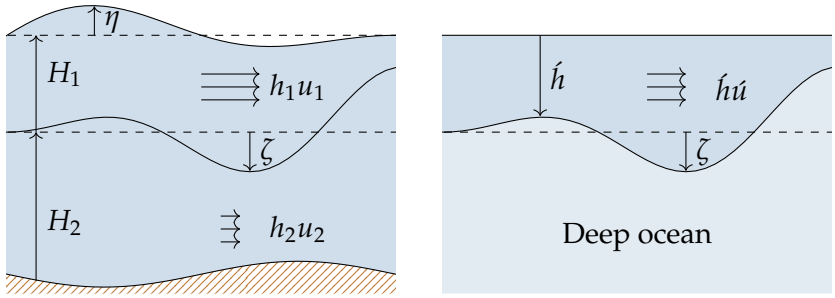


Figure 2.4: GPU OCEAN logo and picture of the GeForce RTX3090 GPU, which was used for most of the experiments in this thesis, taken on 11.09.2023 after it burnt through.

structural patterns to some extent. Moreover, the efficient implementation on GPUs makes the SWE a complementary special-purpose tool for shorter time spans.

In this work, the GPU OCEAN framework has extensively been used for the simulation of simplified forecasts such that a operating GPU set-up was vital, see Figure 2.4. While a NorKyst simulations takes about 1 h 7 min (v2, 512 cores) or 3 h 30 min (v3, 128 cores) on the 'Stratus' cluster at the National Supercomputer Centre, Linköping University, the GPU OCEAN simulation of the same domain and time span takes 1 min 13 s on a GeForce RTX3090, which is an advanced GPU series, or 10 min 48 s on a Quadro T2000, which is a more standard laptop GPU.

Reduced computational time due to accelerated models yields furthermore a cut of financial costs. Considering that financial expenses for the computations are effectuated by number of hours of node usage, more efficient methods reduce the associated price, whereas parallelisation only reduces the wall time but not the total node occupancy. Beyond the financial impact, decreased computational requirements sparse the need for extra hardware resources. This dual effect underscores advantages of efficient computational methods seen from an alternative perspective.



(a) Two-layer shallow-water model **(b)** Baroclinic shallow-water model

Figure 2.5: Schematic cross sections for the two-layer shallow-water and the baroclinic shallow-water model with schematic explanation of the variables.

2.2 Barotropic and Baroclinic Modelling

The integration of the primitive equations over a constant-density water column leads to a pressure depending solely on the depth. As a result, the SWE presented in eq. (2.1) serve as a textbook model for *barotropic* ocean dynamics [113]. This captures, for example, tidal signals or Rossby waves and finds practical applications in scenarios like storm surge forecasting [85]. However, this model ignores *baroclinic* dynamics, which is driven by pressure gradients that are not aligned with the density gradient. Seawater density typically increases with depth and it depends on temperature as well as salinity. The latter quantities vary across space in the ocean leading to varying density profiles for different locations. In particular, protected fjords exhibit a distinct stratification, meaning that a shallow upper layer of lower density ρ_1 and a deep lower layer of higher density ρ_2 are separated by a sharp transition [31].

Determining a mixed-layer depth (MLD), that separates the low density layer on top from the high density region below, a two-layer shallow-water ocean model can be formulated as

$$\begin{bmatrix} h_1 \\ h_1u_1 \\ h_1v_1 \end{bmatrix}_t + \begin{bmatrix} h_1u_1 \\ h_1u_1^2 + \frac{1}{2}gh_1^2 \\ h_1u_1v_1 \end{bmatrix}_x + \begin{bmatrix} h_1v_1 \\ h_1u_1v_1 \\ h_1v_1^2 + \frac{1}{2}gh_1^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ fh_1v_1 \\ -fh_1u_1 \end{bmatrix} + \begin{bmatrix} 0 \\ gh_1H_x \\ gh_1H_y \end{bmatrix} + \begin{bmatrix} 0 \\ -gh_1(h_2)_x \\ -gh_1(h_2)_y \end{bmatrix} \tag{2.7a}$$

$$\begin{bmatrix} h_2 \\ h_2u_2 \\ h_2v_2 \end{bmatrix}_t + \begin{bmatrix} h_2u_2 \\ h_2u_2^2 + \frac{1}{2}gh_2^2 \\ h_2u_2v_2 \end{bmatrix}_x + \begin{bmatrix} h_2v_2 \\ h_2u_2v_2 \\ h_2v_2^2 + \frac{1}{2}gh_2^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ fh_2v_2 \\ -fh_2u_2 \end{bmatrix} + \begin{bmatrix} 0 \\ gh_2H_x \\ gh_2H_y \end{bmatrix} + \begin{bmatrix} 0 \\ -rgh_2(h_1)_x \\ -rgh_2(h_1)_y \end{bmatrix}. \quad (2.7b)$$

This is a coupled system of shallow-water equations for both layers where $r = \rho_2/\rho_1$, see Figure 2.5a, where $H = H_1 + H_2$ and h_1, h_2 are the depth of the upper and lower layer, respectively. However, such a system is no longer unconditionally hyperbolic [21] and the FVM schemes implemented in GPU OCEAN are not sufficient to solve eq. (2.7) on GPUs [35]. Even though the system in eq. (2.7) has the abilities of barotropic as well as baroclinic responses, we follow Røed [114] and continue to isolate the baroclinic response by employing the rigid-lid assumption, meaning that we set $\eta = 0$. By adding the mass conservation equations from eqs. (2.7a) and (2.7b), this requires

$$h_1u_1 = -h_2u_2 \quad \text{and} \quad h_1v_1 = -h_2v_2, \quad (2.8)$$

and the mass conservation reduces to

$$\zeta_t + (h_1u_1)_x + (h_1v_1)_y = 0. \quad (2.9)$$

Lastly, we assume an infinitely deep lower level which implies that we neglect currents in the lower layer, i.e. $u_2, v_2 = 0$ using eq. (2.8). This yields that the equations for the upper layer become decoupled and the momentum equations of the lower layer reduce to

$$\left(\frac{1}{2}gh_2^2\right)_x = gh_2H_x - rgh_2(h_1)_x \quad \text{and} \quad \left(\frac{1}{2}gh_2^2\right)_y = gh_2H_y - rgh_2(h_1)_y. \quad (2.10)$$

With the chain rule, the terms $(h_2)_x$ and $(h_2)_y$ can be extracted and plugged into the force terms of the upper layer.

The resulting so-called 1.5-layer shallow-water model boils down to the same type of mathematics as the barotropic shallow-water model in eq. (2.1). With the baroclinic variables as in Figure 2.5b, the model reads

$$\begin{bmatrix} \zeta \\ \acute{h}\acute{u} \\ \acute{h}\acute{v} \end{bmatrix}_t + \begin{bmatrix} \acute{h}\acute{u} \\ \acute{h}\acute{u}^2 + \frac{1}{2}g\acute{h}^2 \\ \acute{h}\acute{u}\acute{v} \end{bmatrix}_x + \begin{bmatrix} \acute{h}\acute{v} \\ \acute{h}\acute{u}\acute{v} \\ \acute{h}\acute{v}^2 + \frac{1}{2}g\acute{h}^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ f\acute{h}\acute{v} \\ -f\acute{h}\acute{u} \end{bmatrix}, \quad (2.11)$$

where the gravity is reduced according to

$$\acute{g} = \frac{\rho_2 - \rho_1}{\rho_2} g \tag{2.12}$$

and the forcing from the bathymetry cancelled out. Due to eq. (2.12), the system is also called 'reduced-gravity model' and we use it as a simplified model for the baroclinic dynamic in the ocean. Note that we initialise the reduced-gravity model with the baroclinic currents in the mixed layer $\acute{u} = u_1 - u$ and $\acute{v} = v_1 - v$ for consistency, but this is different to the model of Røed [114]. Moreover, the reduced depth of the water column and reduced gravity leads to significantly reduced phase speeds $(\acute{g}\acute{h})^{1/2}$ in the CFL-condition in eq. (2.6), such that larger time steps are achievable and the system can be solved very efficiently in GPU OCEAN.

2.3 Trajectory Modelling

Drift trajectory predictions are, for example, important in SAR missions, but also for collision forecasts of icebergs with offshore facilities or oil spill clean-up. Hereby, the drift properties heavily depend on the shape of the object, how deep the object is submerged into the ocean, and how it orients relative towards the wind. For such incidents in Norwegian waters, trajectories are computed with the OpenDrift software package [30]. Therewith, the forecast outputs from operational models, which commonly have a 1 h time resolution, are read in and trajectories are separately but rapidly simulated using a suitable module in the OpenDrift framework. Such workflows use the ocean models offline, meaning that the currents are not on-the-fly updated according to the latest observations. Drift trajectory prediction based on simplified models enables us to also update the ocean states whenever additional data becomes available.

In GPU OCEAN, the drift modelling is integrated into the SWE simulation, meaning that the drifter locations are updated during the simulation. Drifters are passively advected using the currents u and v interpolated from the relevant grid cells. Additionally, a drift contribution from wind fields can be considered.

Let $\boldsymbol{\psi}^t \in \mathbb{R}^2$ represent the location of a floating object within the computational domain at time t. Further, we denote the bi-linearly interpolated velocities by $u(t, \boldsymbol{\psi}^t)$ and $v(t, \boldsymbol{\psi}^t)$ where the interpolation is defined over the grid cell that contains $\boldsymbol{\psi}^t$ and three of its neighbours. Then, the drifter locations are advanced for the time step Δt_{drift}, commonly chosen to be the

same as the simulator time step Δt, according to

$$\boldsymbol{\psi}_0^{t+\Delta t_{\text{drift}}} = \boldsymbol{\psi}_0^t + \Delta t_{\text{drift}} \left(u(t, \boldsymbol{\psi}^t) + \text{wind}_u(t, \boldsymbol{\psi}^t) \right) \tag{2.13a}$$

$$\boldsymbol{\psi}_1^{t+\Delta t_{\text{drift}}} = \boldsymbol{\psi}_1^t + \Delta t_{\text{drift}} \left(u(t, \boldsymbol{\psi}^t) + \text{wind}_v(t, \boldsymbol{\psi}^t) \right), \tag{2.13b}$$

where $\text{wind}_u, \text{wind}_v$ are contributions from the wind relative to the water velocity. This is not the wind field directly but a function of the wind field.

Chapter 3

Ensemble Forecasts

Numerical simulations are a powerful pillar in the generation of ocean forecasts, but even complex models are only an approximation of reality such that the generated predictions contain inherent uncertainty. To quantify the uncertainty, we consider statistical interpretation from a Bayesian perspective and ensemble prediction systems (EPS) [91, 130, 34]. An ensemble is a set of ocean simulations with slight perturbations and, for example, the Barents-2.5 model of MET Norway is composed of 24 ensemble members [81].

In general, larger ensembles are preferable for more realistic statistical representation, but the ensemble size is limited by available computational resources. Since the SWE are computationally highly efficient, they allow to run ensembles with a bigger number of simulations. This means that we invest the computational gain through the simplification into enhanced statistical power and this is useful for reliable uncertainty quantification.

3.1 Sources of Uncertainty

In the process of mathematical modelling and numerical simulation in Chapter 2, we imposed a series of simplifications and incorporated parameters that are not perfectly known in real-world applications. Likewise, initial conditions and additional forcing terms may be only poorly known. This requires appropriate statistical representation in the forecast system.

To describe the state of a simplified ocean model, we use the *state vector* $\boldsymbol{x} \in \mathbb{R}^{N_X}$. For the SWE model, the state vector collects all the variables over all spatial grid cells at a certain time t, i.e.

$$\boldsymbol{x}(t) = \left(\mathbf{Q}_{j,k}(t)\right)_{j,k=1}^{nx,ny} \tag{3.1}$$

such that $N_X = 3 \cdot nx \cdot ny$. Therewith, the state vector is naturally high-dimensional. The Bayesian paradigm accounts for the inherent uncertainty by prescribing distributions, such that the state vector is interpreted as stochastic process.

Hereby, the uncertainty begins with the initial conditions $\boldsymbol{x}(t^0) = \boldsymbol{x}^0$. The actual state of the ocean at time t^0 is commonly inaccessible or only partially known. Data from operational systems can provide information to set $\boldsymbol{x}^0$ for the SWE, but the operational forecast already includes uncertainty.

For the evolution of the states in time, we embrace the numerical SWE model into the operator $\mathcal{M}^{t^0 \to T}$, which forwards the ocean state from time t^0 to time T. To account for the uncertainty in this temporal evolution, two conceptually distinct statistical modelling approaches can be considered.

First, the simplified models represent only an idealised version of the oceanographic reality and the numerical solution is again only an approximation of the continuous model. To account for unresolved physics, we add an additive error term $\boldsymbol{\epsilon}$ to get

$$\boldsymbol{x}(T) = \mathcal{M}^{t^0 \to T}(\boldsymbol{x}^0) + \boldsymbol{\epsilon}(\omega), \tag{3.2}$$

where ω denotes an element of a sample space. Here, one can include features into the model error that are not captured by the model equations.

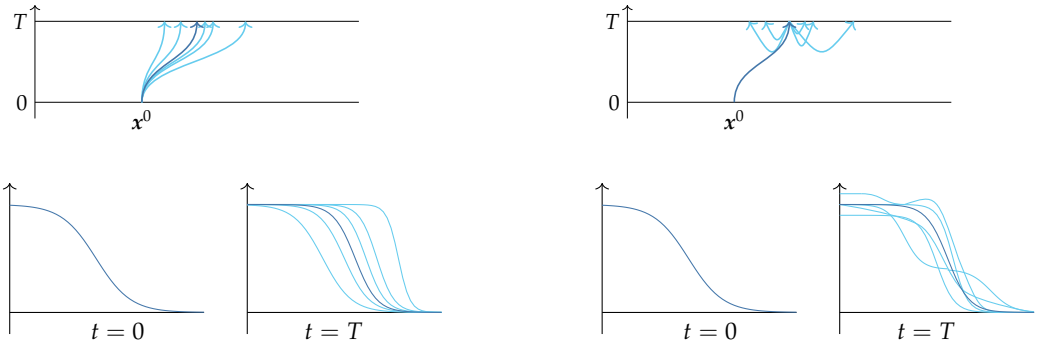
Second, the uncertainty in the model involves the parametrisation of every single forcing term in the problem formulation. Hence, we consider these values as random variables and can sample them from appropriate distributions. This then means that we consider a set of random model parametrisations by

$$\boldsymbol{x}(T) = \mathcal{M}^{t^0 \to T}(\boldsymbol{x}^0, \omega) \tag{3.3}$$

such that the randomness gets directly incorporated into the model operator.

Note that model error is a design choice and should respect realistic conditions, while still being efficient to sample. In particular, we use structures for $\boldsymbol{\epsilon}$ that add random small-scale eddies onto the SWE states, where the eddies are chosen in such a way such that extensive gravity waves are avoided.

The conceptual difference between the model error ansatzes is illustrated in Figure 3.1. The distribution of states with the implicit model error is propagated and transformed through the model equations as visualised in the left display. In the right display, the additive model error determines the distribution of states around the deterministic simulation at time T and



(a) Implicit model error

(b) Additive model error

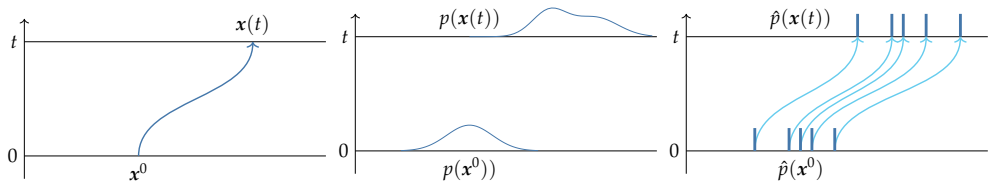
Figure 3.1: The stochastic model error can be incorporated into the model forecasts implicitly or additively. The dark curve represents a deterministic simulation. The upper row depicts the timing when the model error shows effect. The lower row shows an idealised illustration of a travelling wave. The implicit model error perturbs parameters in the model evolution such that all realisations still come from the modelled physics. Conversely, the additive model error allows the inclusion of unresolved features at time T.

its characteristics has to be known a priori. In the lower panels, we also consider a travelling wave to illustrate effects that come with the disparity of the two model error formulations with respect to the introduced features. However, we remark also that the additive model error can be applied with shorter time steps during the simulation from 0 to T. In that case, model error features can develop distinct dynamics over time and the distribution of states at time T is no longer a priori determined. Note that it is also possible to combine both approaches.

3.2 Uncertainty Quantification

In forecast scenarios, it is of interest to quantify the influence of uncertainty represented in the inputs and the model itself onto the final ocean states. While a single deterministic forecast does not provide a comprehensive uncertainty description, a fully probabilistic forecast for the states $\boldsymbol{x}$ with a continuous probability density function (PDF, p) is inaccessible for complex models. If p was available and the following high-dimension integral was solvable, the expected value defined by

$$\mathbb{E}[\boldsymbol{x}] = \int \boldsymbol{x} \cdot p(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} \tag{3.4a}$$



(a) Deterministic forecast **(b)** Probabilistic forecast **(c)** Ensemble forecast

Figure 3.2: Schematic depiction of different forecast types

or the covariance of the state with some random vector $\boldsymbol{y} \in \mathbb{R}^{N_Y}$ defined by

$$\mathbb{C}\mathrm{ov}\left[\boldsymbol{x}, \boldsymbol{y}\right] = \mathbb{E}\left[(\boldsymbol{x} - \mathbb{E}\left[\boldsymbol{x}\right])(\boldsymbol{y} - \mathbb{E}\left[\boldsymbol{y}\right])^\top\right], \tag{3.4b}$$

where also the associated PDF of $\boldsymbol{y}$ is assumed to be known, would be two examples of extracting information from a fully probabilistic forecast.

In contrast to both deterministic and fully probabilistic forecasts, *Monte Carlo* methods rely on repeated sampling of an experiment and are one practical approach to provide forecasts together with the associated statistical uncertainty. In this setting, an *ensemble* is a set of N_e state vectors $(\boldsymbol{x}_e)_{e=1}^{N_e}$, where all ensemble members $\boldsymbol{x}_e$ are independent samples of the same model.

In the Monte Carlo setting, an underlying PDF p is approximated by the ensemble representation and the marginal distribution of $\boldsymbol{x}$ becomes

$$\hat{p}(\boldsymbol{x}) = \sum_{e=1}^{N_e} w_e \delta(\boldsymbol{x} - \boldsymbol{x}_e), \tag{3.5}$$

where $(w_e)_{e=1}^{N_e}$ are weights that sum up to 1 and δ is the Dirac delta function. Starting from uncertain initial conditions x^0 with a given distribution, an ensemble forecast is generated by evolving all ensemble members independently in time. The concept of ensemble-based forecasts in contrast to deterministic and probabilistic forecasts is illustrated in Figure 3.2.

Further, the ensemble members can be used to estimate statistical quantities like the first and second moments at certain times. With equal weights $w_e = 1/N_e$ for all members $e = 1, \ldots, N_e$, the expected value in eq. (3.4a) can be estimated by the sample average

$$\mathrm{E}\left[\boldsymbol{x}\right] = \frac{1}{N_e} \sum_{e=1}^{N_e} \boldsymbol{x}_e, \tag{3.6}$$

which is a vector in $\mathbb{R}^{N_X}$, and using a corresponding ensemble $(\boldsymbol{y}_e)_{e=1}^{N_e}$ with equal weights, the covariance $\mathbb{C}\mathrm{ov}$ in eq. (3.4b) is approximated by

$$\mathrm{Cov}\left[\boldsymbol{x}, \boldsymbol{y}\right] = \frac{1}{N_e - 1} \sum_{e=1}^{N_e} \left(\boldsymbol{x}_e - \mathrm{E}\left[\boldsymbol{x}\right]\right) \left(\boldsymbol{y}_e - \mathrm{E}\left[\boldsymbol{y}\right]\right)^\top, \tag{3.7}$$

which is a matrix in $\mathbb{R}^{N_X \times N_Y}$. The variance $\mathbb{V}\mathrm{ar}\left[\boldsymbol{x}\right]$ is approximated by the diagonal of $\mathrm{Cov}\left[\boldsymbol{x}, \boldsymbol{x}\right]$, which is denoted as $\mathrm{Var}\left[\boldsymbol{x}\right] \in \mathbb{R}^{N_X}$. In the case of arbitrary weights, the mean is estimated by

$$\mathrm{E}\left[\boldsymbol{x}\right] = \sum_{e=1}^{N_e} w_e \boldsymbol{x}_e, \tag{3.8}$$

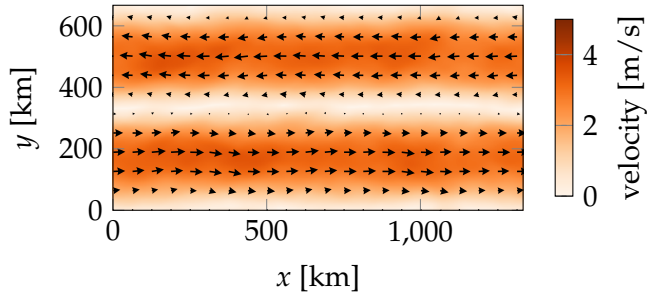
and the covariance of $\boldsymbol{x}$ with itself is estimated by

$$\mathrm{Cov}\left[\boldsymbol{x}, \boldsymbol{x}\right] = \frac{1}{1 - \sum_{e=1}^{N_e} w_e^2} \sum_{e=1}^{N_e} w_e \left(\boldsymbol{x}_e - \mathrm{E}\left[\boldsymbol{x}\right]\right) \left(\boldsymbol{x}_e - \mathrm{E}\left[\boldsymbol{x}\right]\right)^\top. \tag{3.9}$$

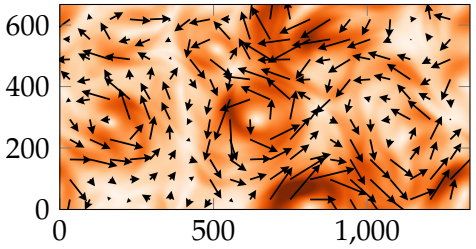
From these estimators for the full state vector, statistics for variables of interest can be derived by selecting the corresponding dimensions. For example for the ocean currents in the SWE model, the statistics can be estimated through its mean $\mathrm{E}\left[u\right], \mathrm{E}\left[v\right]$ and variance $\mathrm{Var}\left[u\right], \mathrm{Var}\left[v\right]$, which are vectors of the same dimension as the computational grid and can be directly visualised.

Monte Carlo estimators are approximations to the true, but inaccessible, expected value and covariance, respectively, and the statistical accuracy of the estimates generally improves for a larger effective number of samples. Hence, larger ensembles with more members are preferable for the quantification of the uncertainty. In ocean forecasting, each Monte Carlo sample corresponds to the simulation of an instance of an ocean model. Since the ensemble size of complex models is limited by the available time budget and computational resources, operational systems provide commonly only a single forecast or a small ensemble. As seen in Section 2.1, the SWE model is computationally more efficient, such that we can afford an increased number of experiments and larger ensembles.

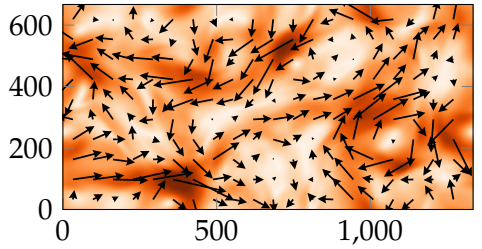
Exemplary, we consider an ensemble of with 500 SWE members for the double jet case from Section 2.1. Each member is initialised by the same initial conditions and evolved by the SWE model for 10 d, where the simulations are perturbed every 1 min with some small additive model error. Then, Figure 3.3 shows the mean velocity in the upper row, meaning $(\mathrm{E}\left[u\right]^2 + \mathrm{E}\left[v\right]^2)^{1/2}$, and velocity fields for two ensemble members in the



(a) Ensemble mean



(b) Ensemble member $e = 1$



(c) Ensemble member $e = 2$

Figure 3.3: Velocities in a ensemble of double jet simulations. The mean is calculated over 500 ensemble members.

lower row. Clearly, the single ensemble members develop turbulent current structures while the mean has a regular pattern reflecting smeared initial conditions.

Ensemble Trajectory Modelling

In our application, the ensemble of ocean states supplies an ensemble of drift trajectories $(\boldsymbol{\psi}_d)_{d=1}^{N_d}$ for some drifter ensemble size N_d. We consider two distinct approaches for the modelling of ensemble trajectories, where both can be integrated into the simulation routine as generally described in Section 2.3. The first one intrudes into every single simulator of the ensemble, while the second one relies on ensemble statistics.

Trajectories by Ensemble Attachment

The most straight forward method for generating an ensemble of drift trajectories from an ensemble of simplified ocean models, is to simulate one

realisation of drift trajectories connected to one ocean model. The drifter locations are updated every time step Δt of the simulator using eq. (2.13) and ensemble of drift trajectories is equally large as the ensemble of SWE models, meaning $N_d = N_e$. The advantage of this is that each individual drift trajectory is consistent with respect to the physics of the ocean model, it is computationally inexpensive, and the statistical distribution of the drift trajectories are in accordance with the distribution of the simulated currents.

Trajectories by Ensemble Statistics

In contrast to the previous approach, the members of the ocean model ensemble are not utilised individually in this alternative. Instead, estimates of the ocean currents are computed and these statistics are used to evolve the drifter ensemble. We sample drifter realisations from a Gaussian process that is centred at $(\mathrm{E}\left[u(t, \boldsymbol{\psi}^t)\right], \mathrm{E}\left[v(t, \boldsymbol{\psi}^t)\right])$ and is scaled by $(\mathrm{Var}\left[u(t, \boldsymbol{\psi}^t))\right], \mathrm{Var}\left[v(t, \boldsymbol{\psi}^t))\right])$. Note that the stochastic realisation spans over the entire simulation time for a sample of this process. In this case, the size of the drifter ensemble N_d can be chosen independent from the SWE ensemble size N_e. To ensure that the realisation of modelled path is consistent over time, we first sample a random component for the currents u and v for every drifter member as

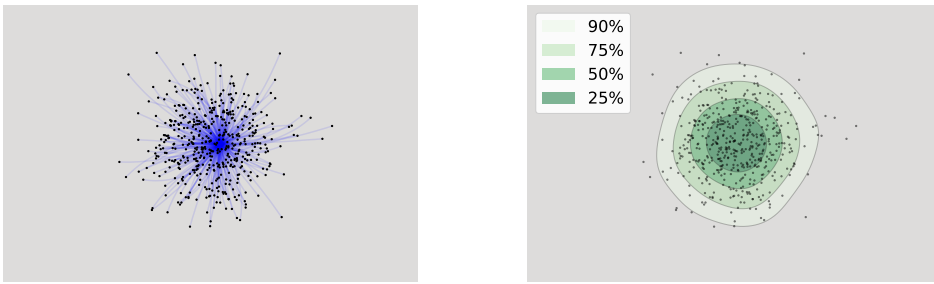
$$\delta_{d,u}, \delta_{d,v} \sim \mathcal{N}(0,1) \quad \text{for } d = 1, \ldots, N_d \tag{3.10}$$

and keep those fixed for the evolution of the drifter in time. The time steps of the drifter evolution Δt_{drift} is also independent of the simulator time steps, but at relevant time steps the means $\mathrm{E}\left[u(t)\right], \mathrm{E}\left[v(t)\right]$ and variances $\mathrm{Var}\left[u(t)\right], \mathrm{Var}\left[v(t)\right]$ are calculated. The drifter ensemble is then forwarded as

$$\boldsymbol{\psi}_{d,0}^{t+\Delta t_{\text{drift}}} = \boldsymbol{\psi}_{d,0}^{t} + \Delta t_{\text{drift}} \left(\mathrm{E}\left[u(t, \boldsymbol{\psi}_d^t)\right] + \delta_{d,u} \sqrt{\mathrm{Var}\left[u(t, \boldsymbol{\psi}_d^t)\right]} + \mathrm{wind}_u(t, \boldsymbol{\psi}^t) \right) \tag{3.11a}$$

$$\boldsymbol{\psi}_{d,1}^{t+\Delta t_{\text{drift}}} = \boldsymbol{\psi}_{d,1}^{t} + \Delta t_{\text{drift}} \left(\mathrm{E}\left[v(t, \boldsymbol{\psi}_d^t)\right] + \delta_{d,v} \sqrt{\mathrm{Var}\left[v(t, \boldsymbol{\psi}_d^t)\right]} + \mathrm{wind}_v(t, \boldsymbol{\psi}^t) \right) \tag{3.11b}$$

for $d = 1, \ldots, N_d$. The currents in eq. (2.13) are replaced by statistical terms. As this approach does not require access to the individual simulations, it is versatile, but assumes intrinsically a Gaussian distribution.



(a) Drifter paths **(b)** KDE

Figure 3.4: Ensemble trajectories and visualisation by drifter paths and KDE

Statistical Modelling of Search Areas

In the Monte Carlo setting for drifters, a point cloud together with the trajectories informs about the paths of the drifter ensemble. However, the discrete representation does not necessarily result in comprehensive visualisations for search areas. As the sampled drifter positions represent an approximation to continuous PDFs, kernel density estimation (KDE) can additionally provide a smooth representation based on the discrete samples at a certain time. For example, for huge ensembles of drift trajectories, a kernel-density presentation is more appropriate than a point cloud. With the estimated covariance of the drifter locations $\boldsymbol{\Sigma}_{\boldsymbol{\psi},\boldsymbol{\psi}} = \mathrm{Cov}\,[\boldsymbol{\psi}, \boldsymbol{\psi}]$ the KDE can be a sum over Gaussian kernels centred at the Monte Carlo samples

$$\hat{p}_{\mathrm{KDE}}(\boldsymbol{\psi}) \propto \sum_{d=1}^{N_d} \exp\left(-\frac{1}{2}(\boldsymbol{\psi} - \boldsymbol{\psi}_d)(\kappa\boldsymbol{\Sigma}_{\boldsymbol{\psi},\boldsymbol{\psi}})^{-1}(\boldsymbol{\psi} - \boldsymbol{\psi}_d)^{\top}\right), \tag{3.12}$$

where κ is the bandwidth and by Scott's rule set as $N_e^{-1/6}$ for a two-dimensional variable [122].

Figure 3.4 pictures the difference between a trajectory plot with drifter paths for every ensemble member from start to end time of the forecast and the KDE of the drifter locations at the final time. The visualised drift trajectory ensemble has 500 members with the same initial position and the black dots mark the drifter locations at the end time of the forecast.

Chapter 4

Data Assimilation

In the previous sections, we have described simplified ocean model and how these can be used to efficiently forecast drift trajectories. Alongside these models and simulations, observation data plays a crucial role in oceanographic forecasts [36]. In this section, we discuss the mathematical framework for how ensembles of numerical models can be updated using such observations through data assimilation.

In general, observations can be obtained from several sources such as satellite images [29], high-frequency radars [126], or buoys [111, 17] and exhibit diverse properties like various resolutions [68]. Similarly to the routines in classical numerical weather prediction, these measurements are utilised in the operational machinery to improve the accuracy of forecasts [98, 107]. However in comparison to the atmosphere, the ocean is only sparsely observed [67]. Operational ocean forecast systems incorporate a wide range of the available observations into the assimilation, but as the models are typically scheduled once or twice a day, the data has to be available before or at least during the run.

In certain time-critical scenarios like SAR missions, additional observations of the real currents can become available during the mission [32]. Such data was not considered in the operational forecasts, as it was likely not yet available during a scheduled run of the computationally intensive models. However, these latest observations can be incorporated into on-demand forecasts with simplified models instead. In this thesis, we consider data from in-situ buoys as these typically take measurements frequently. On the contrary, dozens of kilometres can lie between different buoys, such that the observations are spatially very sparse. Simplified models cannot be expected to resolve all characteristics from general circulation models as there is a natural simplification in the equations and

variables in the SWE compared to the complex models, but an SWE ensemble can benefit from frequent updating according to measurements and can thereby be steered towards the true dynamics.

4.1 The Data Assimilation Problem

To get started, we consider a series of time steps t^n, $n = 0, \dots, N_T$ at which new data is assimilated. Without loss of generality those time steps are assumed to be equidistant. Moreover, we denote the state vector at time t^n as $\boldsymbol{x}^n \in \mathbb{R}^{N_X}$. Starting from an ensemble representation of the initial distribution $(\boldsymbol{x}_e^0)_{e=1}^{N_e}$ and assuming that the ensemble has reached time t^{n-1}, we evolve each member from the previous time step t^{n-1} to the current t^n by

$$\boldsymbol{x}_e^{n,f} = \mathcal{M}^{t^{n-1} \to t^n}(\boldsymbol{x}_e^{n-1}) + \boldsymbol{\epsilon}_e. \tag{4.1}$$

Here, $\mathcal{M}$ represents the numerical SWE model and $\boldsymbol{\epsilon}_e$ is a realisation of the model error as in eq. (3.2). Note that we employ the additive model error as we will actively use its properties later on. Equation (4.1) is called the 'forward step' in the data assimilation cycle and executes for each ensemble member independently. To pronounce that the resulting states are forecasts generated from the physical model, we mark them with the superscript f.

In most geophysical systems, such as in the ocean, states can only be partially observed. In our context, we assume that an *observation* of the (inaccessible) true state $\boldsymbol{x}_{\text{truth}}^n$ is extracted by an operator $\mathcal{H}$ and contains some observation noise. We express the extraction of an observation $\boldsymbol{y}^n \in \mathbb{R}^{N_Y}$ as

$$\boldsymbol{y}^n = \mathcal{H}(\boldsymbol{x}_{\text{truth}}^n) + \boldsymbol{\nu}^n, \tag{4.2}$$

where $\nu^n \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{R})$ is a Gaussian random variable with independent components. Due to the nature of in-situ observations, we restrict the following considerations to linear observation operators $\boldsymbol{H}$ that are matrices of dimensions $N_Y \times N_X$. In particular, if $\boldsymbol{H}$ has only a single 1 per row and 0 otherwise, then this means that $\boldsymbol{H}$ picks the measured state variables at the measurement location. The noise term accounts for various sources of uncertain errors in the data. The measurement tools have error tolerances themselves, and the observations might not directly represent the actual variables in the computational grid. For example, point measurements of currents using buoys do not directly represent the depth and cell averaged values in the discretised SWE model, but they still contain valuable information. As those inaccuracies are often not elsewhere specified, the Gaussian distribution is chosen as generic error law.

A key assumption in the data assimilation framework of this thesis is the sparsity of data in space. Mathematically speaking, we impose

$$N_Y \ll N_X, \tag{4.3}$$

which expresses the big contrast between the high-dimensional state space and the low-dimensional observations.

Model forecasts and available observation data are integrated in *data assimilation* workflows for updating predictions and reducing uncertainty. As this procedure is sequentially repeated whenever new data becomes available, we also speak of *sequential* data assimilation. Bayes' rule provides the formal foundation for data assimilation as

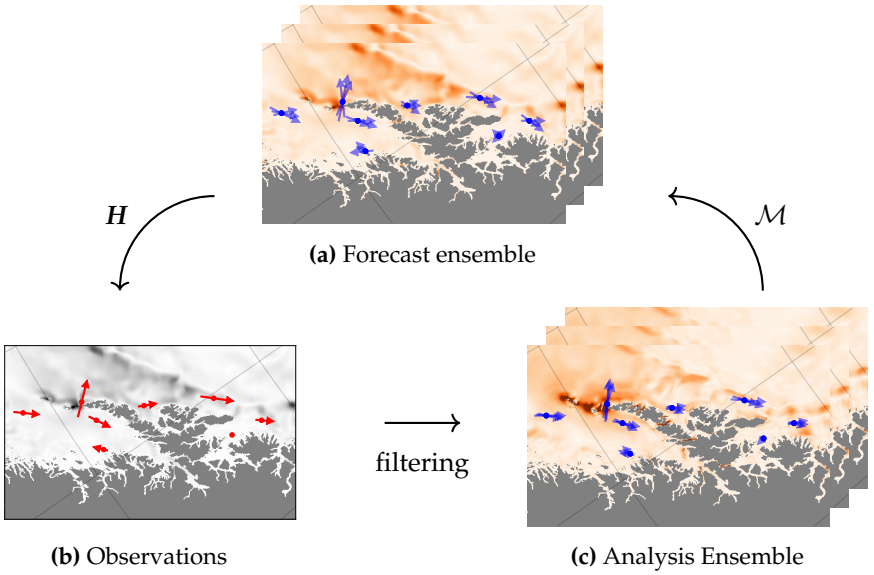
$$p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n}) \propto p(\boldsymbol{y}^n|\boldsymbol{x}^n)p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1}), \quad n = 1, \ldots, N_T. \tag{4.4}$$

starting from the unconditioned prior PDF $p(\boldsymbol{x}^0)$ at t^0. The forecast $p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1})$ that acts as prior is acquired from the forward model before incorporating the latest observation. The likelihood $p(\boldsymbol{y}^n|\boldsymbol{x}^n)$ is the PDF of observing the new data $\boldsymbol{y}^n$ given the model state. For the conditioning, we here use the assumptions of independent error terms in eq. (4.1) and conditionally independent measurements in eq. (4.2). Finally, we are interested in the resulting posterior $p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n})$ or also called filtering PDF that conjuncts the information of prior and likelihood. Together, the prior and the likelihood are the main building blocks for Bayesian inference as Bayes' rule provides the optimal solution of the data assimilation problem, i.e. the posterior, see e.g. [139]. In practice, however, only ensemble approximations of the probability distributions are viable, as we have already acknowledged in Section 3.2. Therefore, Bayes' rule must be interpreted within a Monte Carlo setting.

In contrast to a Monte Carlo setting, ocean forecast systems with a single simulation can only employ variational data assimilation techniques. For this purpose, an optimisation problem is derived from Bayes' rule under the assumption of Gaussian model noise in addition to the Gaussian observation noise. A new state is identified that fits best both the forecast state and the data. In an ensemble version of variational methods, the structures of the covariances in the system are estimated from the samples and go into the optimisation problem, for a review, see [3].

4.2 Ensemble-Based Data Assimilation

In ensemble-based data assimilation, we strive to obtain an ensemble representation of the posterior distribution, and for this task, two major ap-



(a) Forecast ensemble

(b) Observations

(c) Analysis Ensemble

Figure 4.1: Conceptual illustration of the data assimilation cycle.

proaches exist, see e.g. [46, 139, 20]. Particle filters (PFs) aim to solve Bayes' rule directly [137], whereas ensemble Kalman filters (EnKFs) are derived from the analytical solution of Gaussian linear problems [44]. In the following, we will give a short description of each of the two approaches.

Both PFs and EnKFs share the same conceptual cycle that is illustrated in Figure 4.1 for ensemble-based data assimilation. Starting from an initial state, ensemble forecasts are evolved until observations become available. Through filtering methods, an updated ensemble is achieved in one way or another. This is also called the analysis step and we will use the superscript a for the resulting states. Finally, a new forecast step can start.

Particle Filters

PFs are data assimilation methods that are designed to approximate the solution of data assimilation problem through a weighted ensemble. The starting point is typically an ensemble of equiprobable states at time t^0, and throughout the data assimilation cycle, the PF updates the weights according to Bayes' rule. In the basic conception of PFs, the states themselves are not manipulated by the assimilation of observation, such that PFs can be considered to be as fully non-linear data assimilation methods and all states remain consistent with the physics in the forward step at all time.

Starting from a weighted ensemble as in eq. (3.5), representing an approximation to the PDF of the initial state denoted by $\hat{p}(\boldsymbol{x}^0)$ or an approximation to the PDF at the previous model time step denoted by $\hat{p}(\boldsymbol{x}^{n-1}|\boldsymbol{y}^{1:n-1})$, with weights $(w_e^{n-1})_{e=1}^{N_e}$, the forecast PDF approximation becomes

$$\hat{p}(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1}) \propto \sum_{e=1}^{N_e} w_e^{n-1} p(\boldsymbol{x}^n|\boldsymbol{x}_e^{n-1}, \boldsymbol{y}^{1:n-1}). \tag{4.5}$$

To return to a Monte Carlo representation of the forecast, the transition probabilities have to be evaluated in a first step of an algorithm. In the classical bootstrapping approach, this is done by evolving each ensemble member $\boldsymbol{x}_e^{n-1}$ by eq. (4.1) in order to obtain $(\boldsymbol{x}_e^n)_{e=1}^{N_e}$ [59]. In the succeeding update step, the weights are updated according

$$w_e^n \propto p(\boldsymbol{y}^n|\boldsymbol{x}_e^n) w_e^{n-1} \tag{4.6}$$

to approximate the posterior distribution. This assignment updates weights according to the similarity of the observation and the forecast of the ensemble member. Note that only the weights but not the states are manipulated in this purest form of the PF. Hence, we have dropped additional superscripts for the distinction of forecast and analysis states.

The main issue with classical PFs is that over repeated updates, some weights will vanish to almost zero influence, undermining the statistical quality of the ensemble. Hence, resampling approaches are commonly employed, where the goal is to completely disregard low-weight members with negligible statistical contribution, and rather use these computational resources by re-initialising new ensemble members similar to the states of high-weight members [26]. Especially in high-dimensional system like ocean models, it may happen that all states are far away from the observation and a single member carries almost all probability weight. This is called the 'curse of dimensionality' and leads to ensemble degeneration and non-useful results [125]. This is the main reason why PFs are uncommon in ocean data assimilation.

To counteract this behaviour, one could increase the number of ensemble members to postpone the degeneration, but this is rarely computational feasible [75]. Alternatively, one can sample from a proposal density q_e such that the new weights become

$$w_e^{n,*} = \frac{w_e^n}{q_e(\boldsymbol{x}_e^n|\boldsymbol{x}_{1:N_e}^{n-1}, \boldsymbol{y}^n)}. \tag{4.7}$$

The choice for q_e that minimises the variance in the weights is called the optimal proposal and is given by $q_e(\boldsymbol{x}_e^n|\boldsymbol{x}_{1:N_e}^{n-1}, \boldsymbol{y}^n) = p(\boldsymbol{x}^n|\boldsymbol{x}_e^{n-1}, \boldsymbol{y})$, but this

is generally hard to evaluate due to the conditioning on the future data [41]. However, in case of a normally distributed model error $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{Q})$ the optimal proposal remains Gaussian $\mathcal{N}(\boldsymbol{x}_e^{n,*}, \boldsymbol{P})$ with

$$\begin{aligned}\boldsymbol{x}_e^{n,*} &= \mathcal{M}^{t^{n-1}\to t^n}(\boldsymbol{x}_e^{n-1}) \\ &\quad + \boldsymbol{Q}\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1} \left(\boldsymbol{y}^n - \boldsymbol{H}\mathcal{M}^{t^{n-1}\to t^n}(\boldsymbol{x}_e^{n-1})\right) \qquad (4.8a)\\ \boldsymbol{P} &= \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1} \boldsymbol{H}\boldsymbol{Q}, \qquad (4.8b)\end{aligned}$$

where we see that the model error covariance $\boldsymbol{Q}$ is actively used to steer the update ensemble [41].

Since proposal density particle filters may still degenerate in high-dimensional models [103], PF variants that retain equivalent weights have gained attention [135]. The implicit-equal weights particle filter (IEWPF, [145]) is one such example, which keeps all weights equal to a target weight and is unbiased in its two-stage formulation [124]. Interestingly, the IEWPF can be implemented efficiently on the GPU and has been applied to simplified ocean models [76].

Ensemble Kalman Filters

In contrast to PFs that seek to solve the non-linear data assimilation problem directly, EnKFs stem from the linearised data assimilation problem with Gaussian assumptions and the ensemble approximation of its analytical solution. The posterior representation is obtained by updating the states vectors while all weight are kept equal and constant by construction [44].

Following the forward step from eq. (4.1), the EnKF estimates the state covariance matrix from the ensemble and we use the short-hand notation $\boldsymbol{\Sigma}_{x,x}^{n,f} = \mathrm{Cov}\left[\boldsymbol{x}^{n,f}, \boldsymbol{x}^{n,f}\right]$. With this, the *Kalman gain* is constructed as

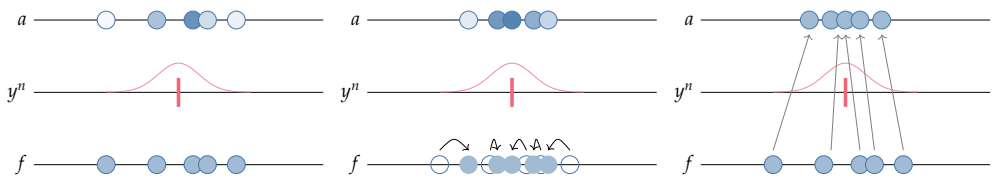
$$\boldsymbol{K} = \boldsymbol{\Sigma}_{x,x}^{n,f}\boldsymbol{H}^\top \left(\boldsymbol{H}\boldsymbol{\Sigma}_{x,x}^{n,f}\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1}. \qquad (4.9)$$

In accordance with the observation model for the true data in eq. (4.2), the forecasted observations from the ensemble are generated as

$$\boldsymbol{y}_e^n = \boldsymbol{H}\boldsymbol{x}_e^{n,f} + \boldsymbol{\nu}_e^n, \qquad (4.10)$$

where $(\boldsymbol{\nu}_e^n)_{e=1}^{N_e}$ are independent realisations of the model error. Finally, the ensemble is updated and the analysis ensemble is obtained by

$$\boldsymbol{x}_e^{n,a} = \boldsymbol{x}_e^{n,f} + \boldsymbol{K}\left(\boldsymbol{y}^n - \boldsymbol{y}_e^n\right), \qquad (4.11)$$



(a) PF **(b)** Optimal Proposal PF **(c)** EnKF

Figure 4.2: Schematic illustration of different ensemble-based data assimilation approaches in one dimension.

see e.g. [19, 77, 47, 136]. Here, the state vectors are manipulated in the update step according to a combination of the Kalman gain, that is the same for all ensemble members, and an individual contribution from the difference between true and forecasted observation. This usually results in a push of the ensemble states towards each other and towards the observation.

First, we note that the EnKF update is similar to the optimal proposal in eq. (4.8a), but the covariance is estimated from the ensemble. Second, for high-dimensional models $\boldsymbol{\Sigma}_{x,x}^{n,f}$ is a large matrix of size $N_X \times N_X$, but its assembling is actually not required. Instead, we can more efficiently use

$$\boldsymbol{\Sigma}_{x,y}^{n,f} = \boldsymbol{\Sigma}_{x,x}^{n,f} \boldsymbol{H}^\top = \mathrm{Cov}\left[\boldsymbol{x}^{n,f}, \boldsymbol{H}\boldsymbol{x}^{n,f}\right], \tag{4.12}$$

which is a matrix in dimensions $N_X \times N_Y$ only. This simplifies the Kalman gain as

$$\boldsymbol{K} = \boldsymbol{\Sigma}_{x,y}^{n,f} (\boldsymbol{H}\boldsymbol{\Sigma}_{x,y}^{n,f} + \boldsymbol{R})^{-1}. \tag{4.13}$$

Even though eq. (4.13) enables more efficient computations [45], further developments as deterministic square-root filter are common, see e.g. [141, 139]. These methods exploit the assumption of a Gaussian posterior and avoid sampling errors due to perturbed model observations. The analysis distribution is defined as $\mathcal{N}(\bar{\boldsymbol{x}}^{n,a}, \boldsymbol{\Sigma}^{n,a})$, which is again derived from the linear Gaussian case. Then, the analysis samples are obtained by

$$\boldsymbol{x}_e^{n,a} = \bar{\boldsymbol{x}}^{n,a} + \tilde{\boldsymbol{x}}_e^{n,a} \tag{4.14}$$

where $\tilde{\boldsymbol{x}}_e^{n,a}$ comes from a matrix square root of $\boldsymbol{\Sigma}^{n,a}$. One such example is the ensemble transform Kalman filter (ETKF, [10]) which works in the ensemble subspace.

A schematic illustration for PFs and EnKF in a single dimension is provided in Figure 4.2. All data assimilation in this subsection methods include in one way or another a forward step yielding a prior ensemble. This

is the equally-weighted dots lying on the lower line depicting the prior states. The displays illustrate how ensembles are re-weighted or moved in the assimilation step. The standard PF only updates the weights, represented by colour shades, while the optimal proposal PF also generates updated states before weighting. In only one dimension, the EnKF update becomes a weighted sum of the prior states and the observation rated by the relation $\frac{1}{1+\sigma/r}$ between the prior variance σ and the observation noise r. The states are moved closer to the observation similar to the update in the optimal proposal PF, but in the EnKF all weights are kept equal.

Localisation

A single observation is mostly valid for describing the conditions in its vicinity, but the limited ensemble size can lead to unintended effects that thwart the filtering update. Therefore, localisation is an essential strategy in practical high-dimensional data assimilation applications.

In the classical PF, we work with weights that are global parameters. So, it may happen that one ensemble member fits well to an observation in one area, but not to an observation in another area. Then, it is difficult to merge different members in such a way that they are a good fit to both observations. One approach lies in sequential processing of observations and smooth reconstruction of posterior ensemble members [110]. In the case of proposal density particle filters, we note that not only weights are changed but also the analysis members are sampled from a new distribution. In particular, this proposal distribution is redefined for each member and incorporates the latest observation data. Thereby, the new ensemble can be steered locally towards each observation.

For EnKFs, there are two common localisation schemes: *local analysis* and *covariance localisation* where in practice the decision for one or the other depends on reasons like computational efficiency [118]. Local analysis follows the paradigm that only data in a certain neighbourhood should influence the update of a grid cell [18]. For each single grid cell or group of grid cells, a subset of close observations is selected and then the update is performed in the local state dimensions only, for example using ETKF [108]. For scenarios with $N_Y \approx N_X$ this scheme is very powerful and can be efficiently parallelised or combined with tapering in the observation influence [80].

Note that the low-rank Monte Carlo estimation of the covariance matrix suffers from spurious correlations that would generate inadvertent long-range updates. In contrast to local analysis, the point of view is reverted

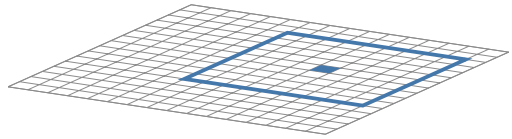


Figure 4.3: Schematic illustration of the interpretation of covariance localisation showing an observation (blue grid cell) together with its area of influence (blue frame). The mesh represents a computational grid for a FVM.

in covariance localisation and the idea is that an observation only affects grid cells that are physically nearby to the observation location, see Figure 4.3. This is done by multiplying the covariance matrix elementwise with a tapering function [78, 79]. As the covariance matrix is often not assembled, this approach can be transferred to the observation space or the Kalman gain [22]. Both localisation approaches can be formulated in such ways that the Kalman gain is not formed in the full state dimensions yielding improved computational efficiency in addition to the truncation of spurious correlations. Herein, we will use Kalman-gain localisation for $N_Y \ll N_X$.

4.3 Assessment of Ensemble Forecasts

Apart from the generation of ensemble forecasts and their refinement by data assimilation, the assessment of the forecast quality is crucial [128, 55]. Therefore, the role of new observational data is extended beyond its use in the assimilation to also serve as reference to assess of the predictive skills. This process where the ensemble forecast is assessed in relation to observations can also be called evaluation or verification [143, Chapter 7] and such considerations enable to evaluate the propriety of the included dynamical models and filtering methods.

Here, we start the discussion for a scalar observation y, meaning that $N_Y = 1$ in eq. (4.2), and we again omit superscripts indicating the time step. An ensemble forecast for the same time, usually after assimilating previously available data, generates synthetic observations y_e as in eq. (4.10). In the Bayesian framework, the observation is associated with some uncertainty and hence it is not favourable when all ensemble members match the observation, but the ensemble should provide a good fit and represent the uncertainty appropriately.

Error metrics like the bias [37] or the root-mean square error [77] are versatile applicable and give good indication of the forecast quality for

single experiments. However, the ultimate ambition is that the synthetic ensemble observations have the same statistical distribution as the real observations [57].

To this end, we recall some statistical foundations for scalar observations and introduce the cumulative distribution function (CDF, F) of a real-valued random variable as

$$F(y) = \int_{-\infty}^{y} p(y^*)\,\mathrm{d}y^* \tag{4.15}$$

where p is the PDF. Then, the CDF takes values from 0 to 1 and is monotonously increasing. A fundamental result states, if a random variable Y is distributed according to F, then

$$F(Y) \sim \mathcal{U}[0,1], \tag{4.16}$$

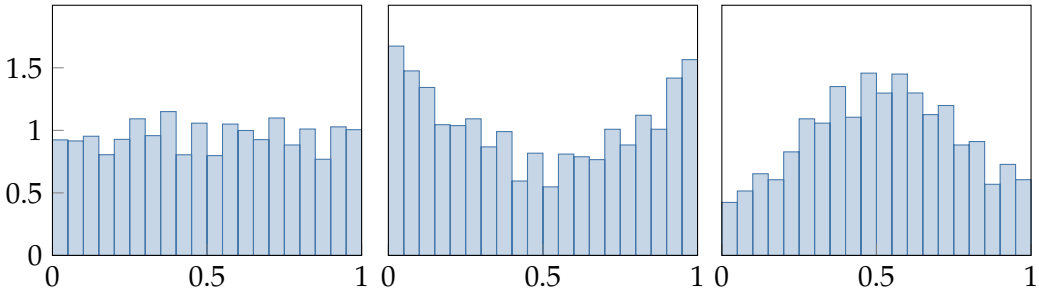
meaning samples of $F(Y)$ are uniformly distributed.

Probability-integral transforms (PIT) are intended to exploit the relation in eq. (4.16) for the assessment of an ensemble forecast. In the Monte Carlo case, the CDF is approximated by the empirical cumulative distribution function (ECDF, $\hat{F}$) which is calculated by eq. (4.15) where the EPDF $\hat{p}$ substitutes the PDF p. The PIT value u is then defined as

$$u = \hat{F}(y), \tag{4.17}$$

meaning that it is the function value of the ECDF for the observation y. Now, one assumes that the observations y follow an unknown statistical distribution Y and if the ensemble represents the same distribution, then the PIT values will be approximative samples from a uniform distribution. For an ensemble with equal weights, the PIT value boils down to counting the rank of the true observation in the sorted ensemble values and normalising by the ensemble size. Therefore, the PIT value is also called 'rank'.

To evaluate whether the observations and the ensemble forecasts have the same distribution, the calculation of PIT values is repeated multiple times for a set of independent observations and ensemble forecasts. This can be done from historical time series or from repetitions of the full experiment. Then, the results are visualised by a histogram and those are popular analysis tool [69]. Due to the result in eq. (4.16), the true distribution will result in flat histograms that are indistinguishable from a uniformly distributed random variable subject to some Monte Carlo noise [43]. Therefore, flat histograms are favourable. Convex shapes indicate an overdispersive ensemble, while a concave shape is an indicator for underdispersion



(a) Calibrated ensemble

(b) Underdispersive ensemble

(c) Overdispersive ensemble

Figure 4.4: Exemplary PIT histograms

and one-sided histograms hint a bias [142, 117]. For an illustrative example, see Figure 4.4.

Beyond histograms, scoring rules reward the conformity of the forecast with the distribution of the observation and can be used to assess the statistical fit of an ensemble [56]. For example, the continuous ranked probability score (CRPS) as proper scoring rule is defined as

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(y^*) - \mathbb{1}_{[y,\infty)}(y^*))^2 \, \mathrm{d}y^*, \tag{4.18}$$

where $\mathbb{1}$ is the indicator function. For the ensemble evaluation, the CDF F can be replaced by the ECDF $\hat{F}$ and the experiment can repeated similarly to the acquisition of PIT values obtaining an average CRPS. If the ensemble and the observation come from the same distribution, then the average CRPS is minimal.

In context of the SWE model, an in-situ observation of the current at a single location in the domain is actually two-dimensional, since currents in both coordinate directions are measured. To use the PIT histograms or CRPS directly, one has to be treated the dimensions individually such that histograms or CRPS are often presented for each variable independently. For the assessment of the joint two-dimensional vector of the current components, there are efforts to elevate the same assessment concepts. Ranks are re-defined to still produce one-dimensional histograms [131] and the CRPS is generalised to the energy score [58].

Chapter 5

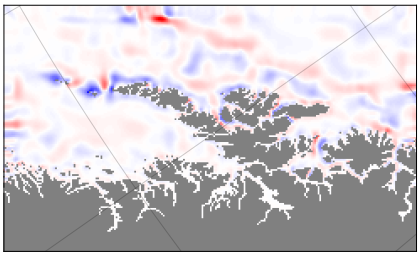
Multi-Level Data Assimilation

In the previous chapters, we have laid the foundation for a complementary ocean forecasting system by introducing simplified ocean models and data assimilation techniques as necessary components. In this chapter, we continue the quest for computational efficiency and aim to save computational costs in the production of forecasts. Therefore, we investigate the computational benefits of using simulations on multiple resolutions and an alternative method to construct the Kalman gain in the data assimilation routine.

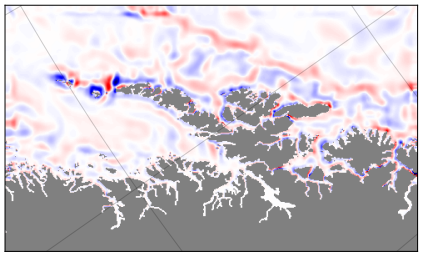
To set the stage, it is helpful to revisit a few properties of the simplified ocean model from Chapter 2: The SWE model offers flexibility in the spatial resolution, where it can also run simulations on a coarser or finer resolution than the operational data. The effect of varying resolution can be inspected, for example, by the vorticity ω_z, which is a measure for the rotation of the water. The rate relative to the earth rotation is defined by

$$\omega_z = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \tag{5.1}$$

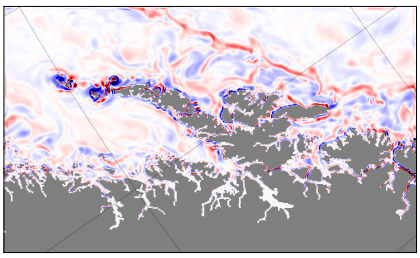
where the notation must not be mixed with the notation for a stochastic realisation. Vorticity can be used to visualise eddies and the level of details in a flow field, and Figure 5.1 shows the vorticity field of simulations with different resolutions in the Lofoten area. All simulations are initialised from the same state extracted from NorKyst and evolved for 3 h. NorKyst runs on the fixed resolution of 800 m, but the SWE in the local area can be run on flexible resolutions [15]. For the SWE simulations, we consider a coarse cell size of 1.600 m, the matching resolution of 800 m, and refined cells with edge length 200 m. In the results on coarse grids, the general patterns of the vorticity are reflected but details are lost. In contrast, the simplified simu-



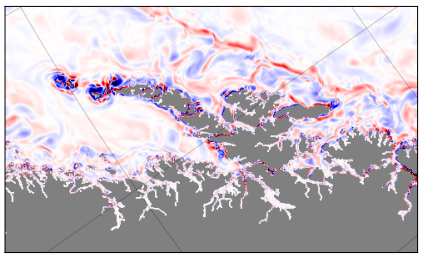
(a) SWE with $\Delta = 1.600\,\mathrm{m}$



(b) SWE with $\Delta = 800\,\mathrm{m}$



(c) SWE with $\Delta = 200\,\mathrm{m}$



(d) NorKyst with $\Delta = 800\,\mathrm{m}$

Figure 5.1: Vorticity after 3 h of simulation with the SWE on different resolutions and with NorKyst on the operational resolution.

lation with the SWE on the finest resolution is very similar to the NorKyst result and keeps the same amount of details as the complex model. The simplified description of the dynamics can be compensated by increased resolution. For this reason, the fine SWE model also stays valid for longer time as the coarse set-ups. Therefore, the ambition is to provide forecasts and conduct data assimilation on the grid with $\Delta = 200\,\mathrm{m}$.

In tandem with the level of detail, the computational run time of the SWE increases towards finer resolutions, whereas coarse grids enable faster simulations but might miss fine-scale details. Nonetheless, those coarse simulations can still resolve large-scale dynamics and can contribute this information inexpensively to statistical estimators. Then, the cost for generation of forecasts can benefit from ensembles of many cheap models on coarse grids for the larger patterns, while more expensive ensembles on fine resolutions fill in features esteeming from small-scale dynamics. The flexible resolution of the SWE facilitates to use information from multiple resolutions at different computational cost per resolution.

In this chapter, we introduce multi-level methods in the data assimilation context. The idea of multi-level methods originates from numerical

integration, where ensembles of different accuracy are combined by a telescoping [70]. Thereby, computational costs can be saved as fewer expensive samples of high accuracy have to be simulated.

This notion extends to statistical estimation such that under certain assumptions a speed-up for the estimation is achieved by preserved statistical quality [52]. In the context of numerical solutions to PDEs, decreasing grid resolutions yield decreasing accuracy by decreasing computational costs. So, the multi-level approach got applied in PDE applications [28, 40] and its potential to reduce computational costs got particularly analysed for hyperbolic conservation laws [100, 96] and the SWE [101]. For data assimilation problems, however, the multi-level concept is less established yet, but it got recently explored for particle filters [82, 61, 90] and ensemble Kalman filters [71, 24].

5.1 Multi-Level Monte Carlo Estimators

Up to this chapter, all members of an ensemble have shared the same computational grid. However, in this chapter, we introduce a hierarchy of grids that cover the same domain, and we use these grids to generate numerical SWE solutions of different accuracy. In abuse of notation, both the grid itself and the grid cell size of the original grid are simply referred to as Δ. We define the hierarchy such that the finest resolution corresponds to the original grid, meaning that we set $\Delta^L = \Delta$. Then, the hierarchy considers additionally a set of grids that are coarser than the original one. The grids in the hierarchy are denoted as Δ^l for $l = 0, \dots, L$ with $\Delta^0 > \dots > \Delta^L$, such that Δ^0 is the coarsest and Δ^L is the finest grid. Note here that the original grid refers to the resolution of interest and should not be mixed with the resolution of the operational grid.

For most of the following considerations, we restrict ourselves to a hierarchy of nested grids, where the domain of a fine cell is covered by exactly one coarser grid cell only. Nested grids can, for example, be generated by iteratively merging neighbouring cells starting from the original grid: two cells in x-direction and two cells in y-direction such that four cells are merged into a coarser one [6]. Such a hierarchy is illustrated in Figure 5.2.

The resolutions in a hierarchy of grids are referred to as *levels*. To distinguish states on different levels, we use the superscript l for the reference to a certain grid level l, and denote states as $\boldsymbol{x}^l \in \mathbb{R}^{N_\mathrm{x}^l}$. Hereby, the states on different levels have distinct dimensions N_x^l according to the varying number of grids cells on each level. Note that we still omit the usage of the time step superscript to prevent confusion.

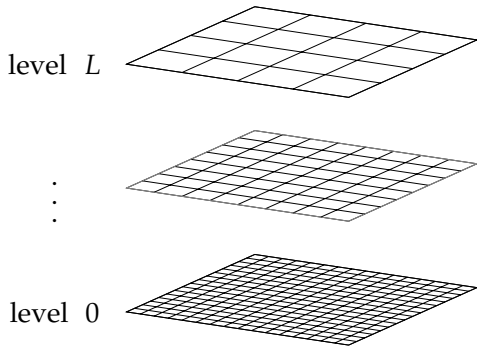


Figure 5.2: Illustration of hierarchy of grids. Original grid with index L on the bottom and lower levels with coarser resolutions above.

In the aforementioned hierarchy of nested grids, the edges size of a grid cell is increased by a factor of two between each level. Then, solving the SWE with FVM on these different levels is associated with varying computational costs and we investigate this here in exemplary manner. Starting from a given level, the number of grid cells on the next coarser grid is reduced by a factor of four. Consequently, the theoretical computational workload to solve the SWE is also diminished by the same factor. This reduction is further amplified by the ability to double the time step size according to the CFL condition in eq. (2.6) which leads to a halving of the number of Runge-Kutta steps. In total, the theoretical cost is significantly reduced by a factor of eight. The computational effort required for the numerical solution of the SWE follows a scaling of $\mathcal{O}(\Delta^{-3})$ in a more general notion [100].

States on all levels essentially describe the same physical quantities in the same domain but on different grids. Then, performing calculations between the vectors of different dimensions necessitates transformations that map a state $\boldsymbol{x}^l$ from level l onto the other levels [48]. For the mathematical description, we denote the projection of a state on level l to a corresponding state on level k as $\boldsymbol{P}_{l\to k} : \mathbb{R}^{N_X^l} \to \mathbb{R}^{N_X^k}$. For a hierarchy of nested grids as before, the projection between two consecutive grids $\boldsymbol{P}_{l\to l-1}$ is a matrix of dimensions $N_X^{l-1} \times N_X^l$ with four $1/4$ per row such that neighbouring values are averaged. Vice versa $\boldsymbol{P}_{l\to l+1}$ is a matrix of dimensions $N_X^{l+1} \times N_X^l$ with four 1 per row such that the same value is repeated on the corresponding finer grid cells. Further projections are generated by repeating these operators. This choice is easy and efficient to implement without explicit construction of these matrices and it preserves global volumes. With these projections, it is possible to execute statistical estimations

on any level. Hereby, the following identities hold

$$\mathrm{E}\left[\boldsymbol{P}_{l\to k}\boldsymbol{x}^l\right] = \boldsymbol{P}_{l\to k}\mathrm{E}\left[\boldsymbol{x}^l\right], \tag{5.2a}$$

$$\mathrm{Cov}\left[\boldsymbol{P}_{l\to k}\boldsymbol{x}^l, \boldsymbol{P}_{l\to k}\boldsymbol{x}^l\right] = \boldsymbol{P}_{l\to k}\mathrm{Cov}\left[\boldsymbol{x}^l, \boldsymbol{x}^l\right]\boldsymbol{P}_{l\to k}^\top. \tag{5.2b}$$

where the transposed projection acts column-wise. This means that there is a direct transformation of states and operations between different levels, such that we omit the explicit notation of projections in the remainder.

Next, we consider two SWE models on two consecutive levels that are initialised with the same data. Note that this includes a projection, but we do not symbolise that. Then, these models will still evolve differently due to numerical errors in the FVM scheme, where we usually trust the simulation on the fine grid more than on the coarse grid. Additionally to sharing the same initial conditions, both instances on the different grids can be perturbed with the same realisation of the model error during the simulation window. This leads to a stochastical coupling between both simulations. Such a pair of simulations is then denoted $(\boldsymbol{x}^{l+}, \boldsymbol{x}^{l-})$, where $\boldsymbol{x}^{l+}$ is defined on the grid Δ^l and $\boldsymbol{x}^{l-}$ on Δ^{l-1}. On the finest level, we however elude this notation and set $\boldsymbol{x}^{L+} = \boldsymbol{x}$ as Δ^L is the original grid.

Having defined states and pairs of states on a hierarchy of grids, we return to statistical estimation. We then exploit a telescoping sum, such that the expected value on the finest level L can be written as

$$\mathbb{E}\left[\boldsymbol{x}\right] = \sum_{l=1}^{L}\left(\mathbb{E}\left[\boldsymbol{x}^l - \boldsymbol{x}^{l-1}\right]\right) + \mathbb{E}\left[\boldsymbol{x}^0\right], \tag{5.3}$$

where the linearity of the expected value was used so that the expected values for all levels $l < L$ cancel out. Here, the telescoping sum acts as a link between the estimation on the finest level and the incorporation of the coarser levels. The analogous formula holds for integrated quantities of interest and motivates the further construction of multi-level estimation methods.

In Section 3.2, the expected value was approximated by the Monte Carlo technique and the use of ensembles. Since all ensemble members in the corresponding estimation in eq. (3.6) are defined on the same grid, we also refer to that classical estimator as a 'single-level estimator'. Moreover, we also use this nomenclature when states first have to be projected onto a common grid, but the same formula is used. Similarly, we continue to approximate the expected values in eq. (5.3). Each of the expected values therein can be estimated by the single-level estimator, but the samples for

the estimation of the expected values $\mathbb{E}[\boldsymbol{x}^l - \boldsymbol{x}^{l-1}]$ are coupled in the sense that the simulations share the same stochastic realisation on both levels. Hence, a *multi-level ensemble* consists of

- a set of states on the coarsest level $(\boldsymbol{x}_e^0)_{e=1}^{N_e^0}$ and
- pairs of states $(\boldsymbol{x}_e^{l+}, \boldsymbol{x}_e^{l-})_{e=1}^{N_e^l}$ on all further levels $l = 1, \ldots, L$,

where N_e^l are the ensemble sizes per level. The corresponding *multi-level estimator* for $\mathbb{E}[\boldsymbol{x}]$ is as an unbiased approximation defined as

$$\mathrm{E}^{\mathrm{ML}}[\boldsymbol{x}] = \mathrm{E}\left[\boldsymbol{x}^0\right] + \sum_{l=1}^{L} \left(\mathrm{E}\left[\boldsymbol{x}^{l+}\right] - \mathrm{E}\left[\boldsymbol{x}^{l-}\right]\right), \tag{5.4}$$

where the single-level estimators are used on the right-hand side. While the single-level Monte Carlo technique originates from an empirical measure as in eq. (3.5), this property is lost for the multi-level Monte Carlo setting. In particular, a Dirac representation associated with the multi-level Monte Carlo approach is in general not a measure, because it can take negative values. Moreover, it is not necessarily bounded by 1, but its approximations of $\mathbb{E}[\boldsymbol{x}]$ have been proven to converge to the correct limit [53].

Beyond the approximation of the expected value, the multi-level formalism is versatile for handling arbitrary quantities of interest. Most importantly for data assimilation, the multi-level approximation of the covariance [8, 105] can be expressed as

$$\mathrm{Cov}^{\mathrm{ML}}[\boldsymbol{x}, \boldsymbol{x}] = \mathrm{Cov}\left[\boldsymbol{x}^0, \boldsymbol{x}^0\right] + \sum_{l=1}^{L} \left(\mathrm{Cov}\left[\boldsymbol{x}^{l+}, \boldsymbol{x}^{l+}\right] - \mathrm{Cov}\left[\boldsymbol{x}^{l-}, \boldsymbol{x}^{l-}\right]\right), \tag{5.5}$$

where again the single-level estimators from eq. (3.7) are recycled. Unlike in the classical single-level estimation, there is no guarantee that $\mathrm{Cov}^{\mathrm{ML}}[\boldsymbol{x}, \boldsymbol{x}]$ is a positive semi-definite matrix.

5.2 Practicalities for Multi-Level Ensembles

Having the formalism for estimators based on multi-level ensembles at hand, we shift our attention more towards the practical implementation. We return to the intention of achieving computational benefits by this procedure, where we aim to retain the statistical quality of the estimator and simultaneously reduce computational costs. Therefore, we start with

analysing the computational costs and the associated statistical error, which consequently enables us to choose appropriate ensemble sizes on each level.

To conduct the subsequent discussion slightly more general, we introduce the quantity of interest $\mathbb{E}[g(\boldsymbol{x})]$ with $g : \mathbb{R}^{N_X} \to \mathbb{R}^{N_g}$ for some N_g. The expected value from eq. (5.4) is then recovered with g equal to the identity mapping. While this notation makes the examination applicable to a larger range of quantities of interest, it also exhibits that the results in the following are intrinsically dependent on g.

Computational Costs

Computational costs manifest in the wall time that it takes to evolve a SWE simulation forward in time on available hardware. We usually simulate until a certain time where we want to conduct an estimation such that this corresponds, in other words, to the generation of the samples for the estimation. The multi-level approach gains advantage from coarse grids, where the SWE are computationally less expensive to solve than on the original grid. We keep all other factors like schemes for FVM fluxes and time integration fixed, such that the computational cost is solely determined by the grid resolution. The computational cost associated with solving the SWE on level 0 are denoted as C^0 and the costs for a pair $(\boldsymbol{x}^{l+}, \boldsymbol{x}^{l-})$ are represented as C^l, where C^l for $l > 0$ actually entails the numerical solution on two grids.

As motivated already in the previous section, the costs scale in theory as $\mathcal{O}(\Delta^{-3})$ over the grids. In practice, the scaling of the computational costs can however be influenced by technical factors. In ocean models, parallelisation is a common strategy to accelerate the run times of simulations. Both parallelisation and multi-level methods are primarily independent efforts for reduced run times, but they follow different pathways and can interfere in certain regards. Here, parallelisation leads typically to reduced computational costs at full scale of the problem, but for very small grid sizes that do not utilise the entire parallelism of the implementation the theoretical scaling may no longer hold true.

Statistical Error

After evolving simulations up to certain time, we consider the accuracy of the statistical estimation itself. The statistical error in the estimation τ_g is defined as the variance of the estimator for $\mathbb{E}[g(\boldsymbol{x})]$ and is given for a

single-level ensemble by

$$\tau_g^2 = \frac{\| \mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x})\right] \|}{N_e}, \tag{5.6}$$

where $\| \cdot \|$ is a norm over $\mathbb{R}^{N_g}$. Both the ensemble size and also the quantity of interest g itself influence the statistical quality. While $\mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x})\right]$ is unknown in practice and has to been estimated itself, the variance term is determined by the inherent distribution of the states and the the function g. Notwithstanding, an increase in ensemble size can reduce the statistical error. In the classical single-level context, the choice of the ensemble size is typically dictated by the available computational resources and it is simply accepted that, e.g. mean and covariance are estimated with different statistical errors.

For a multi-level ensemble, the statistical error for the estimation is analogously described by

$$(\tau_g^{\mathrm{ML}})^2 = \frac{\| \mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x}^0)\right] \|}{N_e^0} + \sum_{l=1}^{L} \frac{\| \mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x}^l) - g(\boldsymbol{x}^{l-1})\right] \|}{N_e^l}, \tag{5.7}$$

where the exact variances are again inaccessible in practise. Moreover, to achieve a small statistical error for the multi-level estimator, the identity

$$\begin{aligned} &\mathbb{C}\mathrm{ov}\left[g(\boldsymbol{x}^l) - g(\boldsymbol{x}^{l-1}), g(\boldsymbol{x}^l) - g(\boldsymbol{x}^{l-1})\right] \\ &\qquad = \mathbb{C}\mathrm{ov}\left[g(\boldsymbol{x}^l), g(\boldsymbol{x}^l)\right] + \mathbb{C}\mathrm{ov}\left[g(\boldsymbol{x}^{l-1}), g(\boldsymbol{x}^{l-1})\right] \\ &\qquad\qquad - 2\mathbb{C}\mathrm{ov}\left[g(\boldsymbol{x}^l), g(\boldsymbol{x}^{l-1})\right] \end{aligned} \tag{5.8}$$

suggests that the levels should be strongly correlated. This correlation is a key indicator of scenarios where the multi-level approach can actually aid computational efficiency. Subsequently, we use the notations $V^0 = \| \mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x}^0)\right] \|$ and $V^l = \| \mathbb{V}\mathrm{ar}\left[g(\boldsymbol{x}^l) - g(\boldsymbol{x}^{l-1})\right] \|$ for $l > 0$.

Multi-Level Ensemble Size

A key design choice for the implementation of multi-level ensembles is the assignment of the ensemble sizes N_e^l on each level. These ensemble sizes are determined by solving an optimisation problem that minimises the computational costs under the constraint of fixed statistical error for the estimation of the quantity of interest [104]. Based on this, we obtain

$$N_e^l = \left\lceil \sqrt{\frac{V^l}{C^l}} C_\tau \right\rceil \quad \text{with } C_\tau = \frac{1}{\tau_g^2} \sum_{l=0}^{L} \sqrt{V^l C^l}, \tag{5.9}$$

where τ_g is in our case given by the statistical error of a reference single-level ensemble. In relevant conditions, the ensemble sizes will typically increase towards the coarser levels [53].

Beyond the definition herein, it is possible to construct best linear unbiased multi-level estimators where the state pairs extend to more than two levels, both for the scalar case [121] and for the multi-variate case [38]. The multi-level estimation is then reformulated as a linear regression problem and weights are introduced for the coupled simulations. The weights are chosen to minimise the variance of the multi-level estimator further. While it is conceptually possible to integrate this into the preceding definitions, the coupling of more than two levels introduces further computational overhead in the forward step due to the exchange of the perturbation realisations between the different simulation instances such that we restrict our considerations to the case with two coupled levels.

Computational Speed-Up

To evaluate the computational effect of the multi-level approach, we need to examine the total computational costs of the different types of ensemble simulations. In the case of a single-level ensemble, the total costs are obtained from the product of ensemble members N_e and the computational costs c^L for a single simulation on the original level. This amounts to $\mathcal{C} = N_e \cdot c^L$. In the context of multi-level estimation, the costs are aggregated over the levels and summarised as

$$\mathcal{C}^{\mathrm{ML}} = \sum_{l=0}^{L} N_e^l \cdot C^l. \tag{5.10}$$

For ensembles that yield the same statistical error the computational impact is quantified as the speed-up

$$\mathcal{S} = \frac{\mathcal{C}}{\mathcal{C}^{\mathrm{ML}}}. \tag{5.11}$$

An acceleration is indicated if $\mathcal{S} > 1$ and the speed-up provides a measure of how much faster the multi-level ensemble is compared to the single-level ensemble or vice versa how much slower the single-level ensemble is compared to the multi-level ensemble.

Albeit, the multi-level approach harvests the computational efficiency of coarse grids, it will not always yield a speed-up. The potential of multi-level estimation depends crucially on the coupling between the levels. Strong coupling on fine levels in relation to the general variance in the system and

steep computational scaling are favourable for a speed-up. If the system we want to simulate is too turbulent on the small scale, meaning that the correlation between levels is weak, we cannot expects any benefits from using coarser levels. The opposite situation can also be imagined, where the correlations between levels are so strong such that there are effectively no extra details to be found on the finest level compared to a coarser level. In that case, the theoretical speed-up is large but the finest simulations do not add any value to the results, such that in practice it would be better to neglect fine levels or to use simply a single-level ensemble on a coarser resolution.

Application to Simplified Ocean Models

As seen in Figure 5.1, the SWE model necessitates increased spatial resolutions for high-quality forecasts. While the computational costs for simulating the SWE grow in accordance with the resolution, the model can be easily run on different resolutions in a hierarchy of nested grids. Moreover, the interpretation of the vorticity suggests that the levels are, to a visual extend, correlated, such that the simplified ocean model appears to be suited for the application of multi-level methods. In this case, for instance, we would be interested in constructing a multi-level SWE ensemble with $\Delta = 200\,\mathrm{m}$ as resolution of interest.

Multi-level estimators enable to gather statistical information from a multi-level SWE ensemble on the original grid. For example, mean velocity and variance can be calculated at certain times. These estimates therefore facilitate a seamless incorporation of a multi-level ensemble for drift prediction according to the trajectory modelling by ensemble statistics in Section 3.2, where the conventional estimators for mean and variance are replaced by their multi-level analogons in eq. (3.11). This promotes the usability of multi-level ensembles for the generation of drift trajectory forecasts for SAR scenarios.

5.3 Multi-Level Ensemble Kalman Filter

Transitioning from multi-level Monte Carlo methods without conditioning on observation data towards multi-level data assimilation (MLDA), we build upon the EnKF method outlined in Section 4.2. The multi-level ensemble Kalman filter (MLEnKF) applies multi-level estimation to the Kalman gain [71] and has been extended by a version for spatio-temporal data assimilation problems [24]. This approach has also been combined

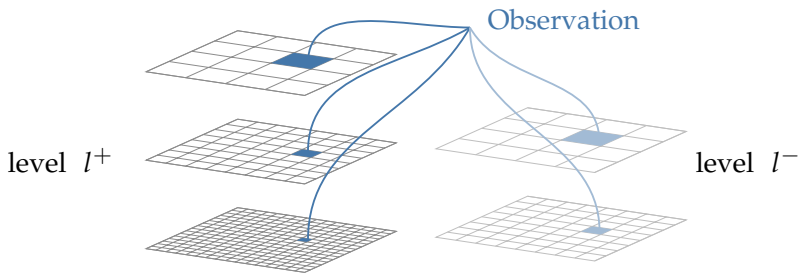


Figure 5.3: Illustration of the observation operations in a multi-level ensemble

with Bayesian model averaging (see e.g. [112]) and applied for reservoir simulations [50, 106].

In the MLDA context, a multi-level ensemble of SWEs is evolved in time until data becomes available. The observation model in eq. (4.2) means then that the data is associated with the finest level. This implies that coarser states have to be upscaled to the finest level before the observation operator is applied. For point observations in the SWE model, this is still equivalent to picking the current values from the grid cell that contains the observation location, see Figure 5.3. Therefore, we introduce the shorthand notation $\boldsymbol{H}^l$ for extracting data from level l.

For the assimilation of observation data into the multi-level ensemble, we adapt the method of Chernov et al. [24] and apply multi-level estimation for the Kalman gain, but phrase it for nested grids as introduced above. In the construction of the Kalman gain, the estimate of the state-observation covariances in eq. (4.12) is replaced with an estimate as in eq. (5.5). At the n-th time step of the sequential data assimilation cycle, we obtain

$$\begin{aligned}\boldsymbol{\Sigma}_{\boldsymbol{x},\boldsymbol{y}}^{n,f,\mathrm{ML}} &= \mathrm{Cov}\left[\boldsymbol{x}^{0,n,f}, \boldsymbol{H}^0\boldsymbol{x}^{0,n,f}\right] \\ &\quad + \sum_{l=1}^{L}\left(\mathrm{Cov}\left[\boldsymbol{x}^{l+,n,f}, \boldsymbol{H}^l\boldsymbol{x}^{l+,n,f}\right] - \mathrm{Cov}\left[\boldsymbol{x}^{l-,n,f}, \boldsymbol{H}^{l-1}\boldsymbol{x}^{l-,n,f}\right]\right). \quad (5.12)\end{aligned}$$

This multi-level covariance estimate is plugged in to eq. (4.13) and the resulting Kalman gain, denoted by $\boldsymbol{K}^{\mathrm{ML}}$, has the same dimensions as a single-level Kalman on the finest level.

In the update step, all levels are conditioned using $\boldsymbol{K}^{\mathrm{ML}}$. To do so, the spatial dimension of the Kalman gain is downscaled onto the respective level and the analysis multi-level ensemble is given on the coarsest level

by

$$\boldsymbol{x}_e^{0,n,a} = \boldsymbol{x}_e^{0,n,f} + \boldsymbol{K}^{\mathrm{ML}} \left(\boldsymbol{y}^n - \boldsymbol{H}^0 \boldsymbol{x}_e^{0,n,f} - \boldsymbol{\nu}_e^{0,n} \right) \tag{5.13a}$$

and on the higher levels by

$$\boldsymbol{x}_e^{l+,n,a} = \boldsymbol{x}_e^{l+,n,f} + \boldsymbol{K}^{\mathrm{ML}} \left(\boldsymbol{y}^n - \boldsymbol{H}^l \boldsymbol{x}_e^{l+,n,f} - \boldsymbol{\nu}_e^{l,n} \right) \tag{5.13b}$$

$$\boldsymbol{x}_e^{l-,n,a} = \boldsymbol{x}_e^{l-,n,f} + \boldsymbol{K}^{\mathrm{ML}} \left(\boldsymbol{y}^n - \boldsymbol{H}^{l-1} \boldsymbol{x}_e^{l-,n,f} - \boldsymbol{\nu}_e^{l,n} \right), \tag{5.13c}$$

where pairs share the same observation noise $\boldsymbol{\nu}_e^{l,n}$, recall eq. (4.10).

The MLEnKF framework enables to combine multi-level ensembles in the forward model and EnKF-based data assimilation. Hereby, a computational speed-up for MLDA can originate from exploiting cost efficient coarser levels in the forward step. While the updating with MLEnKF shares the basic concept with the corresponding single-level data assimilation workflow, the forecast and analysis ensemble are represented on multiple levels. On the one hand, MLEnKF inherits the computational benefits of multi-level Monte Carlo methods in the forward step, but on the other hand it also takes over the shortcomings. In particular, the multi-level ensemble size in a MLDA workflow is fixed, where the optimal configuration is usually chosen for the single estimation of a certain quantity of interest, but the ensemble is used for repeated estimations of multiple quantities of interest. Also negative eigenvalues in the covariance estimation gain new significance since these covariances are used in the update process and thereby corrupted estimations during one update step could be transported into the further evolution of the ensemble.

5.4 Assessment of Multi-Level Ensemble Forecasts

Analogously to the Section 4.3, there arises the need to assess multi-level ensemble forecasts originating from a MLDA workflow with respect to observations. For error metrics like the bias and the root-mean square error, the multi-level estimation framework as in eq. (5.4) can be employed with a relevant quantity of interest. Beyond those, we emphasise the evaluation whether forecast and observation distributions are in accordance. As for single-level ensembles, the assessment of the calibration of multi-level ensemble forecasts with the traditional means relies on approximations of the EPDF or ECDF. However, those are less straight forward to determine for multi-level ensembles, since there is no associated measure [54, 86].

Here, we give a short survey of selected approaches for the assessment of multi-level ensemble forecasts. In the case of scalar observations, Gregory and Cotter [62] suggest to resample a single-level ensemble such that the classical evaluation tools can be employed. Hereby, the resampled ensemble is a set of new scalar values $(y_e^L)_{e=1}^{N_r}$ for $N_r \leq N^0 + \sum_{l=1}^{L}(N^l - 1)$ with dedicated analytical weights. With the resampled single-level ensemble, PIT histograms or the average CRPS can be calculated as in Section 4.3.

Alternative ways, that also work in two dimensions, take advantage of multi-level estimation. Given an observation y, the identity $u = \mathbb{E}[\mathbb{1}_{(-\infty,y]}(y^*)]$ can re-define the PIT value and more generally also the CDF F, when y^* is distributed according to F. Then, the expected value can be approximated by the multi-level estimator. However, the multi-level ensemble size is not optimised for this quantity of interest and due to the discontinuity the approximation converges poorly [42]. This leads to artefacts like a non-monotonous ECDF and PIT values outside the interval between 0 and 1. Therefore, the indicator function can be replaced with kernel-based smoothing to diminish artefacts [129].

Moreover, a polynomial approximation to the PDF can be calculated based on Legendre polynomials and the maximisation of the Shannon entropy [9]. In doing so, the coefficients in the optimisation problem are set using the multi-level estimation. The numerical optimisation involves iterative solvers that are not guaranteed to converge, but when converged the resulting ECDF is monotonously increasing from 0 to 1. With approximations of the CDF, it becomes possible to re-use PIT histograms and the CRPS from Section 4.3 also for multi-level ensembles.

Chapter 6

Paper Contributions

The path towards improved local ocean and drift forecasting is genuinely interdisciplinary. Therefore, the papers in this thesis traverse through different scientific fields. While each paper focuses on a particular problem, they all contribute to the same goal and are thereby closely connected. This is a typical example that developments in applied computational statistics require to link knowledge from several mathematical disciplines to solve the problem at hand.

The papers in this thesis cover multiple aspects in the data assimilation value chain. Starting from the design of a simplified ocean model and its numerical simulation, via a filtering case study, to state-of-the-art assimilation with multi-level ensembles, the papers explore the suitability of advanced methods for their application in local ocean drift forecasting systems by proof-of-concepts. Hereby, each paper is dedicated to one the research objectives from Chapter 1.

Paper I Considerations for the FVM that is used solve the SWE. Field of numerical mathematics.

Paper II Simplified ocean modelling with the SWE respecting different dynamical modes for efficient drift trajectory forecasting. Field of mathematical modelling for EPS in oceanography.

Paper III In-depth comparison of two conceptually different filters and discussion of localisation strategies tailored for sparse observations in simplified oceanographic applications. Field of data assimilation.

Paper IV Combination of the multi-level approach from the field of uncertainty quantification with practical EnKF-based data assimilation.

Open-Source Software

Reproducibility is a key principle in scientific research and refers to the practice of making methodologies, data, and computational workflows openly available so that others can verify and replicate the findings. In computational science, open-source code plays a vital roles in enabling reproducible science, advancing scientific knowledge, and promoting transparency and collaboration [93, 94]. As all papers in this thesis rely essentially on numerical and statistical implementations, all supplementary material is available in an open persistent repository alongside with the manuscripts to foster to the reproduction and extension of the presented results. The source code and data is published under the GNU Public Licence version 3.

Paper I

Reducing Numerical Artifacts by Sacrificing Well-Balance for Rotating Shallow-Water Flow

Håvard Heitlo Holm, Florian Beiser

Finite Volumes for Complex Applications X—Volume 2, Hyperbolic and Related Problems

DOI: 10.1007/978-3-031-40860-1_19

This contribution was presented by Håvard at the FVCA10 conference 2023 in Strasbourg (France).

The CDKLM scheme is well-balanced for geostrophic jets along the coordinate axes. To achieve this, it employs a reconstruction based on potential energies with respect to geostrophic imbalance and a standard-upwind (SU) flux in one of the vector components instead of the central-upwind (CU) flux as in the other components, recall Section 2.1. In this paper, we show that this choice of fluxes can lead to oscillatory artefacts in cases that differ from geostrophic jets. In particular, by considering an independent implementation of KP, we confirm that the main source of this numerical behaviour is the flux choice and not the reconstruction. Therefore, we propose to use a weighted sum of SU and CU flux instead of the pure SU flux to suppress the unintended artefacts.

For the testing of the scheme, we consider five cases that are relevant for oceanographic applications. We present that already a small fraction of CU is able to reduce oscillations. On the contrary, the well-balance guarantee is lost, but we note that this guarantee for CDKLM anyways only holds for the jets and not for other types of geostrophic steady states. However, for suitable choices of the slope limiter parameter in the reconstruction step, we see that even a pure CU flux preserves geostrophic jets within acceptable tolerances.

While FVM schemes generally have few tuning parameters, we have hereby introduced one more for the weighting of the flux terms. This comes at the advantage of more realistic simulations, though, and based on a test case with challenging topography and currents, we suggest to use a fraction of 0.8 for the SU flux and 0.2 for the CU flux.

Contribution

While the development of generally well-balanced schemes is still ongoing [39], specific FVMs are only well-balanced for particular steady states.

We demonstrate that a FVM that is well-balanced for a superset of steady states compared to another gives more numerical artefacts when applied to other general problems. This shows that the use of a simpler flux expression in favour for the theoretical results is not fruitful for practical applications. With these results, this paper supports the discussion around the choice of numerical solvers for realistic situations where no scheme is well-balanced for all general steady-states.

In particular, the CDKLM scheme for the rotational SWE is well-balanced for a superset of geostrophic steady states compared to the KP scheme, but it replaces a CU with a SU flux. By weighting both flux formulations, we have adapted numerical simulations for simplified ocean model to produce more realistic results.

Further Work

The development of numerical FVM schemes is not the core of this work, but is a necessary ingredient on the way to the full workflow. Therefore, it will be relevant to follow the advances in the development of new well-balanced schemes.

Comments

This paper tackles a problem that have been observed in some debugging cases, but for a long time the cause of these artefacts was not found. Finally, in this paper, we identify the issue and present a pragmatic strategy for how to mitigate these oscillations where we sacrifice the well-balance in favour for the reduction of numerical artefacts. The test cases are taken from previous publications [73, 15] and we see generally only minor improvements compared to the overall flow patterns what implies that their results stay qualitatively valid.

Xing et al. [144] note that well-balanced schemes whose equilibrium is closest to the present case should be used. However, this becomes difficult in practice. Geostrophically balanced states are central in many real-world applications, nonetheless, we still believe that improved realistic simulations are more important than the well-balanced guarantee in academic test cases as long as the loss of well-balance in those cases is within acceptable limits.

Paper II

Combining Barotropic and Baroclinic Simplified Models for Drift Trajectory Predictions

Florian Beiser, Håvard Heitlo Holm, Martin Lilleeng Sætra, Nils Melsom Kristensen, Kai Håkon Christensen

In review

A summary of this paper was presented in the October 2023 instalment of Ocean Modelling Seminar organised by Tor Nordam in SINTEF Ocean under the title 'Combining simplified models in GPU OCEAN*'. The seminar series takes place in Trondheim, but is broadcast online to the interested audience.*

In this study, we place emphasis on simplified ocean models as lightweight decision-support tool for ensemble predictions of drift trajectories. We exploit the SWE and the GPU OCEAN software for the efficient simulation of barotropic and baroclinic dynamics independently, i.e. we use the reduced-gravity model as a simplified model for first-order baroclinic dynamics as outlined in Section 2.2. The choice of using either the barotropic or baroclinic model or both is determined based on the specific local conditions in the area of interest, and the existence of a well-defined mixed layer. We describe how the same structure of equations is re-used and how the system can be parameterised. Given the uncertainties, when complex effects are parameterised and several parameters are not precisely known in realistic scenarios, we sample these randomly from appropriate probability distributions and analyse sensitivities. This means that the model error here is implicit, see eq. (3.3). Yet, the reduced-gravity model can be simulated faster in relation to the barotropic SWE, such that we can afford a large ensembles to represent the uncertainty adequately. By combining the currents from both simplified models, we construct large ensembles of drift trajectories for short-term prediction. As proof-of-concept, we showcase the feasibility of our approach for three distinct regions along the Norwegian coast and generate drift ensembles of up to 27,000 members.

Contribution

In this paper, we promote simplified ocean models as complementary tool. While operational ocean models prioritise physical complexity and become computationally expensive, we explore how lightweight simplified ocean models can be used for ensemble predictions and uncertainty representation. We demonstrate the fast prediction of drift trajectory ensembles using a computationally efficient framework.

In particular, the existing SWE simulation framework is expanded by the reduced-gravity model for respecting simplified baroclinic dynamics. Moreover, we present a concept to efficiently sample a large number of drift trajectories by the combination of simplified models for barotropic and baroclinic modes that pushes the ensemble size to new magnitudes. The efficient generation of large sample sets is fundamental for ensemble-based data assimilation and therewith this work can be seen as enabler for the combination of drift simulations and filtering.

Further Work

To leverage the light-weight computations even further, one could take advantage of non-linear data assimilation methods. Here, parameter choices that turn out to be a good representations of the true dynamics acquire larger weights, while those parameters that fit poorly are rejected. Thereby, the predictive power will be increased.

Moreover, the initialisation of models and ensembles is a well-known challenge in the oceanography community. For example, the definition of meaningful MLDs is non-trivial, but has substantial influence on the results. There is always potential for improving the initial conditions and the models that parameterise, e.g. wind forcing and friction. This will become more relevant when validating the ensemble drift trajectories against real drifter data.

Comments

While we categorised this work first as low-hanging fruit, the reality of ocean states crossed our assessment. The proper initialisation and parameterisation is essential, i.e. we have seen that the MLD and the baroclinic currents as integrated from three-dimensional models may not be geostrophically balanced. Therefore, the reduced-gravity model is easier to initialise in local areas such that this approach is mostly suited for local drift trajectory forecasting, which is in line with our original interests, though. We see significantly improved drift trajectories in the fjords, when also baroclinic dynamics is considered. Moreover, for drift trajectories, the wind acts twofold: on the drifting object itself and the currents in the upper ocean layer. The baroclinic model responds faster to winds than the barotropic model and hereby increases the realism. In a nutshell, this methodology can be useful as one applies the simplified framework and interprets the result with care. Furthermore, the generation of drift trajectory ensembles has direct implication for SAR cases.

Paper III

Comparison of Ensemble-Based Data Assimilation Methods for Sparse Oceanographic Data

Florian Beiser, Håvard Heitlo Holm, Jo Eidsvik

Quarterly Journal of the Royal Meteorological Society

DOI: 10.1002/qj.4637

Early versions of the content of this paper were presented at NORDSTAT 2021 in Tromsø (Norway) under the title 'Comparison of ensemble-based data assimilation methods for drift trajectory forecasting' and at the EnKF Workshop 2022 in Balestrand (Norway) under the title 'Handling sparse observations in ensemble-based filtering with an application to drift trajectory forecasting'.

In this paper, we focus on spatially very sparse in-situ observations and compare two ensemble-based data assimilation methods, namely the IEWPF and the ETKF with localisation. We elaborate on the handling of sparse observations $N_y \ll N_x$ and localisation in the context of both methods. We go step-by-step through the update of the IEWPF and describe how these concepts are built into the IEWPF by design. Hereby, all properties of the IEWPF are tightly dependent on the structure of the model error covariance matrix $\boldsymbol{Q}$. Moreover, we explore what happens if different $\boldsymbol{Q}$ matrices are used in the optimal proposal. For the ETKF, we employ covariance localisation in observation space. Together with serial processing of observations, this gives us good control of the correlations between observations and of the computational costs. We calculate the updates only for grid cells within the localisation radius, and since the Kalman gain is never explicitly constructed in the ETKF, we apply tapering in the final analysis computation step.

As a test case, we use the double jet where the experiment is set up with 450,000 state variables and 120 observation variables. For the comparison of the data assimilation methods, we investigate a broad range of evaluation metrics: quantitative skill scores, qualitative spatial state estimation, and drift trajectory forecasting. First of all, we emphasise that a set of metrics, rather than a single one, should always be used when comparing different methods to gain a deeper insight. Then, we conclude that the IEWPF does not degenerate in this application, but introduces artefacts originating from the model error covariance matrix into the ensemble spread. In the ETKF with localisation, we utilised relaxation-to-prior that is dependent on a tuning factor. Upon proper tuning, the ETKF with localisation outperformed the IEWPF in all metrics.

Contribution

This paper provides the first extensive and structured comparison of the IEWPF with an established version of the EnKF respecting the special scenario of sparse observations. Moreover, we present an elaborate set of metrics to gain insight in the filtering results of two conceptually different methods.

In particular, we dwell on tailored techniques for sparse in-situ buoy observations and on characteristics of the assimilation step itself in the context of the SWE model.

Further Work

Even though this study is rather self-contained, parts of the results open up for a discussion on parameter optimisation in the filter. In particular, the parametrisation of the covariance structures in IEWPF could be a topic of further analytical considerations. From a practical perspective, the parameters in the localisation and relaxation could be more interesting.

Comments

The setting of this paper is motivated by the conditions during SAR operations, i.e. only simplified ocean models are feasible due to their computational efficiency and only sparse in-situ observations are available for assimilation. In this work, we tailor data assimilation methods to these conditions and this study is therefore relevant for the identification of suitable filters. The findings herein support our choice to utilise the EnKF with the employed localisation scheme in our next article.

While Holm et al. [76] focused on GPU-acceleration, we have shifted our focus to the comparison with the ETKF and sacrificed the GPU discussion in this paper. In addition to the SWE model, we also include a linear Gaussian example that provides an analytical reference solution. This example allows us to verify our implementations, analyse the impact of tuning parameters in the localisation, and emphasise the importance of using a comprehensive set of evaluation metrics.

Paper IV

Multi-Level Data Assimilation for Simplified Ocean Models

Florian Beiser, Håvard Heitlo Holm, Kjetil Olsen Lye, Jo Eidsvik

In review

The presentation at the ISDA conference 2023 in Bologna (Italy) with the title 'Rank Histogram Estimators for Multi-level Data Assimilation' originated from discussions around this article.

We pursue the gain of computational efficiency in the data assimilation cycle for the EnKF by employing multi-level methods. Therefore, we commence by revising the MLEnKF for the GPU OCEAN framework. We put special effort into the discussion of the configuration of the multi-level ensemble and of the practical applicability of the MLEnKF in a relevant setting with simplified ocean models. To that end, we investigate the discrepancy between theoretical and practical scaling of computational costs, analyse the coupling between the levels, and their effects for the multi-level speed-up. This means that we determine trial experiments to set the values for computational costs C^l and the variances V^l from Section 5.2. While the choice of the quantity of interest, i.e. the function g, emerges as a central design choice for the configuration of a multi-level ensemble, we disclose that data assimilation routines necessitate the estimation of various quantities of interest. Moreover, we incorporate localisation into the Kalman gain construction and a mitigation strategy for negative eigenvalues in the estimated covariance matrices.

We resume to the double jet case, but in contrast to Paper III we use a finer resolution and a Karhunen-Loeve-type model error. On a hierarchy of four nested grid levels, the coarsest grid clearly lacks details that are resolved on the finer grids. We set up three multi-level configurations that share the same theoretical statistical error for a quantity of interest as a given single-level ensemble, but achieve a theoretical speed-up of up to more than a factor 2 in the forward step. Numerical experiments for the double jet case confirm that the state estimation with the MLEnKF is of the same quality as with the corresponding single-level EnKF while the spread of the multi-level ensemble is slightly higher. Finally, we present the usability of the multi-level ensemble for drift trajectory forecasting by exploiting ensemble statistics, see Section 3.2.

Contribution

Multi-level Monte Carlo estimation is an established tool in the uncertainty quantification community, but the application of the multi-level idea to spatio-temporal data assimilation problems is so far still limited. With this paper, we elevate the MLEnKF one step from analytical test cases to application-driven proof-of-concept studies. Moreover, we provide an extensive discussion around the construction of suited multi-level ensembles.

In particular, we design multi-level data assimilation for simplified ocean models, where the forward step exploits simulations on different resolutions, and investigate the applicability of the MLEnKF for the shallow-water equations with sparse observations.

Further Work

This work can be extended in several directions. Our example was only one step towards more realistic settings, and we had periodic boundary conditions and a flat sea bed. To continue this direction, the consistent incorporation of a varying bathymetry and land mask will become relevant. Hereby, special care has to be dedicated again to the coupling between the levels such that artificial noise is avoided.

The calculation of computational costs is purely based on the forward step, while the update step is assumed to be of negligible cost for all methods. For a naive implementation of the MLEnKF, we have observed that it takes longer than a naive implementation of the single-level EnKF. Naturally, it could be worth to examine this effect and do the technical labour to tune the implementation of the update step itself as the total run time matters.

In addition, we have discussed that suboptimal scaling of the GPU OCEAN simulator on very coarse grids diminishes the speed-up through the use of multi-level methods. Also the multi-level ensemble benefits from the fast computation on the fine levels, but the potential of MLDA could be increased when approaches for the acceleration of coarse levels are inspected. While the computational costs of small problem sizes is usually not relevant in the development of ocean modelling software, it matters for multi-level methods.

Furthermore, it is conceptually not necessary to use the same model equation on all levels. While keeping the variables projectable and the perturbations transferable, the simplified models in our study could be extended by levels $l > L$ from more complex models. In such multi-fidelity hierarchies [64], the increased accuracy of the results does not only come

from increased resolution but also from increased physical complexity.

Comments

While we keep structurally the same data assimilation method as in Paper III, the multi-level method focuses on accelerating the data assimilation cycle by a speed-up in the forward step. As computational efficiency is a key for drift predictions in SAR scenarios, reducing computational time for the generation of trajectories is beneficial.

With the formulation of the MLEnKF, we stay close to the formulation of Chernov et al. [24], but we use practical evaluation metrics as we do not restrict ourselves to analytical cases. Also Fossum et al. [50] employ multi-level data assimilation in a practically relevant case, but we do not decouple the higher levels by hybridisation with Bayesian model averaging in our notion. Still, it is remarkable how little the MLEnKF results are sensitive to the evaluated configurations. Moreover, in the composition of the paper, the design of the model error became a discussion in itself, since it has to be communicated between two levels each and has to be appropriately parameterised.

Finally, the application of MLDA is only beneficial in very specific conditions. On the one hand, the correlation between the finest levels has to be sufficiently strong to obtain a speed-up, and on the other hand, there is no need to refine if the fine resolution does not contribute with increased detail information. Hereby, the double jet is actually a representative case and it is challenging for multi-level estimators since the dynamics between the jets becomes turbulent with many small-scale details.

Chapter 7

Concluding Remarks and Outlook

In this thesis, we have pursued a complementary approach for local drift forecasting motivated by search-and-rescue scenarios. Rather than relying on a single prediction from a complex model, we have employed ensembles of simplified ocean models to represent the probability distribution of the currents and to assimilate observation data from in-situ measurements into the forecasts. The computational efficiency of the shallow-water model and the flexibility of the numerical solvers have facilitated the generation of large ensembles on different resolutions and multi-level data assimilation has enabled to refine the statistics for high-resolution forecasts.

Numerical simulations of simplified ocean models can be nested locally into operational forecasts and predict drift trajectories on demand. Here, we have utilised the shallow-water equations for the simplified modelling of barotropic and, in the presence of a distinct mixed-layer, also baroclinic dynamics.

In a Monte Carlo approach, we used ensembles of simulations to represent uncertainty in the forecast arising, among others, from the initial states and the dynamical model. Furthermore, this approach has taken benefit of assimilating measurements that had not been considered in the operational workflow. While we assume that observational data is sparse in time-critical applications, we have analysed tailored data assimilation techniques that steer the ensemble forecasts towards the observations and generally improve the forecast skills.

Finally, as computational efficiency plays a crucial role in search-and-rescue operations, we have explored ensembles on a hierarchy of levels

and their potential for further computational speed-ups. Therefore, we have used the multi-level ensemble Kalman filter to assimilate observational data into shallow-water ensembles.

7.1 Concluding Remarks

Seen in the bigger picture, the combination of simplified models and data assimilation enables rapid forecasts with flexible resolutions and uncertainty information in local areas, making this approach a well-suited addition for search-and-rescue scenarios. The work in this thesis constitutes a proof-of-concept and an in-depth study of the methods in the development chain towards a complementary system for drift trajectory forecasting.

While simplified methods do not aim to replace the operational forecasting systems, the concepts in this thesis represent novel pathways and build a base for further steps to establish the combination of simplified models and data assimilation as on-demand tool. In the frame of developing such a complementary system, this thesis provides scientific contributions for the proof-of-concept of the involved dynamical model and data assimilation techniques. Also in the scientific context, our considerations are motivated by the application in mind and the results show the value, but also limitations, of shallow-water models and multi-level data assimilation as a special-purpose ensemble prediction system for drift trajectory forecasting.

For the simplified ocean models, we have confronted well-balanced finite-volume methods and purely barotropic models with realistic conditions accentuating the need for pragmatic solutions. In the finite-volume scheme, we have discussed the trading of well-balance in favour for reducing numerical artefacts in classical oceanographic cases and have improved the results for general applications. Moreover, we have investigated an extension of the barotropic shallow-water model by the reduced-gravity models and have manifested the efficient generation of large ensembles of drift trajectories.

Continuing with ensemble-based data assimilation, we have compared two conceptually different filtering methods for the handling of sparse observations. Here, we have evaluated a recent particle filter and a version of the ensemble Kalman filter, such that we have gained further insights into their implicit or explicit localisation. In this assessment, we exhibit that the use of advanced non-linear data assimilation methods does not always lead to better forecast statistics than linear updating methods.

Last, we have brought multi-level data assimilation one step ahead and have addressed practical challenges on the way. We have presented the applicability of multi-level ensemble and the associated potential for computational speed-ups for challenging data assimilation problems. Here, we have succeeded to apply the ensemble Kalman filter for simplified ocean models.

From my personal perspective, as if our project leader Jan Erik Stiansen would ask for an appraisal of the research in this thesis, the general approach of simplified models and multi-level data assimilation can find its place in future ensemble prediction systems for drift trajectories. While offline drift models will stay faster and complex models will stay more precise, ensembles of simplified models fit in between. The advantages of efficient forecasts with realistic uncertainty representation and the assimilation of observations that would have otherwise been ignored are prevalent.

However, to add value to trajectory predictions, all methods have to be adequate. While the barotropic shallow-water model is hereby rarely sufficient for drifter modelling in protected fjords, the supplementary reduced-gravity model enables even larger ensembles, but its robust initialisation in complex conditions remains a challenge for further research.

We have identified suitable data assimilation methods for sparse in-situ observations. Moreover, we have incorporated multi-level ensembles into the architecture, where the theoretical computational cost was halved in an exemplary scenario with the shallow-water equations. Transferring the acquired knowledge to the broader perspective of a complementary forecast system for search-and-rescue systems, the practical impact of multi-level data assimilation for local high-resolution ensemble forecasts depends on multiple factors. The computational scaling is one aspect and, for parallel code infrastructures, this depends on the number of grids cell on the level of interest. While the restriction to local domains casts scepticism whether the massive parallelisation in the GPU OCEAN simulation framework will be fully occupied, yielding suboptimal scaling, the high resolution of interest indicates big problem sizes and allures the utilisation of cost efficient coarser grids. In addition, the correlation of the levels in relation to the ensemble variance on the coarsest level indicates whether one can expect a reduction of computational costs in the ensemble simulation compared to single-level ensembles. These variances are still unknown for real-word scenarios, but could be subject of further investigations.

7.2 Outlook

For practical impact on search-and-rescue operations at sea, it would be necessary to elevate the discussed methods onto a robust operational level with support for all potentially occurring real-world conditions. This would involve a lot of technical work, but also gives rise to further scientific investigations.

As the complex ocean models and data assimilation frameworks will develop in future, those will also become computationally more efficient. However, if simplified models with rapid data assimilation ripens further, they can find their place as complementary first-response systems in the operational architecture. As mentioned in Chapter 6 for each individual paper, there are several directions to expand the research. Most prominently for the technology readiness of the general approach are developments for respecting real-world conditions. Beyond this, multi-resolution simulations, transfer to other application scenarios or machine learning forecasts are interesting topics.

Real-world Conditions

The most natural extension would be to take the methods from this thesis to real-world domains. As the SWE model already accommodates for real-world domains and topographies, the challenges will lie in the ensemble initialisation and generating a realistic ensemble spread through out the simulation.

The initial conditions are not perfectly known and the ensemble is supposed to properly account for this uncertainty. However, for instance in the case of NorKyst, only a single forecast is available such that it requires innovative ways to initialise an ensemble that represents a realistic uncertainty. Moreover, we have used two different types of model errors in this thesis for temporal perturbations of the ocean states, but for both types it remains an open challenge to create a realistic ensemble spread respecting the actual uncertainty in the dynamics [49]. These facts demand the analysis of real-world data compared to the simplified models and identification of suitable perturbation designs, this can, for instance, include machine learning techniques for error corrections [140].

The aforementioned steps will be relevant to both single-level and multi-level ensemble methods. For the multi-level case also further work, e.g. for mismatching land masks on different levels, see Chapter 6, is needed and the variances between the levels should be analysed to evaluate the potential for a computational speed-up through the use of multi-level methods.

Thereafter, it will be interesting to benchmark the approach of simplified ocean models and data assimilation against operational drift trajectory forecasts and real drifter data.

Assessment of Multi-Level Ensemble Forecasts

With real-world data, the proper assessment of ensemble forecast will become pivotal. For the evaluation of multi-level ensemble forecasts with respect to observational data by conventional techniques, like probability integral transform histograms or scoring rule, multiple approaches have been presented in Chapter 5. Hereby, all methods involve preparatory steps, such as resampling, smoothing, or polynomial approximation, before the actual assessment. Consequently, it could be interesting to study the potential effect of these methods on the validation outcomes in more detail.

Multi-resolution Models

In Paper IV, we have employed global coarsening in our examples, but it may be prohibitive to coarsen in the full domain as too many features are lost, e.g. in some coastal or turbulent areas, whilst in other areas coarsening is still feasible. Therefore, it could be relevant to keep high-resolution in some areas and coarsen only in others [84].

Vice versa, the ambition can be to even increase the resolution in specific areas of interest [116]. High-resolution data of the topography may be easier accessible than corresponding forecasts for the ocean currents on such a fine grid. Hereby, challenges will occur, for example, when initialising currents around small islands that have not been resolved in the operational data. For ensembles of multi-resolution models, the same locally refined grid could be used for all ensemble members or a hierarchy of such grids could supply a multi-level ensemble. Accordingly, either the single-level or the multi-level data assimilation methods from this thesis could be applied. In contrast, super resolution data assimilation [4] follows an alternative approach and simulates only coarse ensemble members globally, but uses neural nets to fill in high-resolution features. Here it can be interesting to compare both approaches in the end.

Combined Barotropic and Baroclinic Simplified Models

Foremost, the initialisation of the reduced-gravity model could be further explored. Such simplified approaches do not have the ambition to be a

perfect model of the real dynamics, but enhanced by observational data they could be advanced to special-purpose tools for drift trajectory prediction. Therefore, the impact of data assimilation on combined barotropic and baroclinic simplified models could be analysed. For details, see Chapter 6.

Other Applications

Search-and-rescue operations have motivated the setting of this thesis, but the developed methods are not restricted to this. Storm surges are dominantly a barotropic phenomenon, such that the barotropic shallow-water model could be re-used for storm surge modelling and rapid sea level rise forecasting [85]. In addition to current observations, measurements of the sea level would be a relevant data source for assimilation in such an application [138]. The framework could be extended to other current-driven tracers as long as the interest lies in local short-term trajectories. For ice berg drift [23], the use of an ensemble of simplified ocean model together with stochastic drift and forcing parameters could represent the uncertainty in the driving currents better than a single deterministic current field with corresponding parameter samples. Observations in the arctic would rather originate from satellite pictures than from buoys. Furthermore, supplementary models, for instance for feed or waste in fish farming, and data collection types could be coupled with the shallow-water model and the ensemble Kalman filter. Also in other societal and economical relevant contexts, such a combination of models can be applied. For example, the predictions for the spread of an acute algae bloom require efficient models and fast updates [120]. In all cases, the data assimilation techniques would need to be adapted to the specific application. This may include multi-level formulations of other versions of the EnKF, when the data is no longer sparse.

Machine Learning Models

Machine learning models are an emerging technology also in typical data assimilation applications, like meteorology or oceanography. After an extensive training period, these models produce partial or full forecasts very fast as they circumvent to solve the physical equations directly. These methods can either be used in combination with classical data assimilation [13, 12] or replace the traditional workflow completely as presented very recently for weather forecasting [109, 89, 7, 83]. During all further development on simplified models for drift trajectory forecasting, it could

be reasonable to keep an eye on the progress of machine learning models whether those can be a useful and practical enhancement or supplement.

Bibliography

[1] J. Albretsen and L. Asplin. Hvilken betydning har oppløsning for kyst-og fjordmodeller?-Validering og representasjonsberegninger av strømmodeller med eksempler fra Sulafjorden, Møre og Romsdal. Technical report, Institute of Marine Research, 2021.

[2] J. Albretsen, A. K. Sperrevik, A. Staalstrøm, A. D. Sandvik, F. Vikebø, and L. Asplin. NorKyst-800 Report No. 1 User Manual and Technical Descriptions. *Fisken og Havet*, 2:1–48, 2011.

[3] R. N. Bannister. A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):607–633, 2017.

[4] S. Barthélémy, J. Brajard, L. Bertino, and F. Counillon. Super-resolution data assimilation. *Ocean Dynamics*, 72(8):661–678, 2022.

[5] M. Bentsen, I. Bethke, J. B. Debernard, T. Iversen, A. Kirkevåg, Ø. Seland, H. Drange, C. Roelandt, I. A. Seierstad, C. Hoose, et al. The Norwegian earth system model, NorESM1-M–Part 1: Description and basic evaluation of the physical climate. *Geoscientific Model Development*, 6(3):687–720, 2013.

[6] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3):484–512, 1984.

[7] K. Bi, L. Xie, H. Zhang, X. Chen, X. Gu, and Q. Tian. Pangu-weather: A 3D high-resolution model for fast and accurate global weather forecast. *arXiv preprint arXiv:2211.02556*, 2022.

[8] C. Bierig and A. Chernov. Convergence analysis of multilevel Monte Carlo variance estimators and application for random obstacle problems. *Numerische Mathematik*, 130(4):579–613, 2014.

[9] C. Bierig and A. Chernov. Approximation of probability density functions by the multilevel monte carlo maximum entropy method. *Journal of Computational Physics*, 314:661–681, 2016.

[10] C. H. Bishop, B. J. Etherton, and S. J. Majumdar. Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly weather review*, 129(3):420–436, 2001.

[11] R. Bleck. An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modelling*, 4(1):55 – 88, 2002. doi: 10.1016/S1463-5003(01)00012-9.

[12] M. Bocquet. Surrogate modeling for the climate sciences dynamics with machine learning and data assimilation. *Frontiers in Applied Mathematics and Statistics*, 9:1133226, 2023.

[13] J. Brajard, A. Carrassi, M. Bocquet, and L. Bertino. Combining data assimilation and machine learning to infer unresolved scale parametrization. *Philosophical Transactions of the Royal Society A*, 379:20200086, 2021.

[14] A. Brodtkorb. *Scientific Computing on Heterogeneous Architectures*. PhD thesis, University of Oslo, 2010.

[15] A. Brodtkorb and H. Holm. Coastal ocean forecasting on the gpu using a two-dimensional finite-volume scheme. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 73(1):1–22, 2021.

[16] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, and O. O. Storaasli. State-of-the-art in heterogeneous computing. *Scientific Programming*, 18(1):1–33, 2010.

[17] U. Brönner, M. Sonnewald, and M. Visbeck. Digital twins of the ocean can foster a sustainable blue economy in a protected marine environment. *The International Hydrographic Review*, 29(1):26–40, 2023.

[18] K. Brusdal, J.-M. Brankart, G. Halberstadt, G. Evensen, P. Brasseur, P. J. van Leeuwen, E. Dombrowsky, and J. Verron. A demonstration of ensemble-based assimilation methods with a layered OGCM from the perspective of operational ocean forecasting systems. *Journal of Marine Systems*, 40:253–289, 2003.

[19] G. Burgers, P. J. Van Leeuwen, and G. Evensen. Analysis scheme in the ensemble Kalman filter. *Monthly weather review*, 126(6):1719–1724, 1998.

[20] A. Carrassi, M. Bocquet, L. Bertino, and G. Evensen. Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5):e535, 2018.

[21] M. Castro-Díaz, E. Fernández-Nieto, J. González-Vida, and C. Parés-Madroñal. Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system. *Journal of Scientific Computing*, 48(1):16–40, 2011.

[22] Y. Chen and D. S. Oliver. Localization and regularization for iterative ensemble smoothers. *Computational Geosciences*, 21(1):13–30, 2017.

[23] S. Cheng, A. Aydoğdu, P. Rampal, A. Carrassi, and L. Bertino. Probabilistic forecasts of sea ice trajectories in the Arctic: Impact of uncertainties in surface wind and ice cohesion. *Oceans*, 1(4):326–342, 2020. doi: 10.3390/oceans1040022.

[24] A. Chernov, H. Hoel, K. Law, F. Nobile, and R. Tempone. Multilevel ensemble Kalman filtering for spatio-temporal processes. *Numerische Mathematik*, 147(1):71–125, 2021. doi: 10.1007/s00211-020-01159-3.

[25] A. Chertock, M. Dudzinski, A. Kurganov, and M. Lukácová-Medvidová. Well-balanced schemes for the shallow water equations with Coriolis forces. *Numer. Math.*, 138(4):939–973, 2017. doi: 10.1007/s00211-017-0928-0.

[26] N. Chopin, O. Papaspiliopoulos, et al. *An introduction to sequential Monte Carlo*, volume 4. Springer, 2020.

[27] K. H. Christensen, Ø. Breivik, K.-F. Dagestad, J. Röhrs, and B. Ward. Short-term predictions of oceanic drift. *Oceanography*, 31(3):59–67, 2018.

[28] K. A. Cliffe, M. B. Giles, R. Scheichl, and A. L. Teckentrup. Multilevel Monte Carlo methods and applications to elliptic PDEs with random coefficients. *Computing and Visualization in Science*, 14:3–15, 2011.

[29] J. Cummings, L. Bertino, P. Brasseur, I. Fukumori, M. Kamachi, M. J. Martin, K. Mogensen, P. Oke, C. E. Testut, J. Verron, et al. Ocean data assimilation systems for GODAE. *Oceanography*, 22(3):96–109, 2009.

[30] K.-F. Dagestad, J. Röhrs, Ø. Breivik, and B. Ådlandsvik. OpenDrift v1.0: A generic framework for trajectory modelling. *Geoscientific Model Development*, 11(4):1405–1420, 2018.

[31] S. B. Dalsøren, J. Albretsen, and L. Asplin. New validation method for hydrodynamic fjord models applied in the Hardangerfjord, Norway. *Estuarine, Coastal and Shelf Science*, 246:107028, 2020.

[32] F. J. Davidson, A. Allen, G. B. Brassington, Ø. Breivik, P. Daniel, M. Kamachi, S. Sato, B. King, F. Lefevre, M. Sutton, et al. Applications of GODAE ocean current forecasts to search and rescue and ship routing. *Oceanography*, 22(3):176–181, 2009.

[33] H. Davies. A lateral boundary formulation for multi-level prediction models. *Quarterly Journal of the Royal Meteorological Society*, 102(432):405–418, 1976.

[34] V. de Aguiar, J. Röhrs, A. M. Johansson, and T. Eltoft. Assessing ocean ensemble drift predictions by comparison with observed oil slicks. *Frontiers in Marine Science*, 10:1122192, 2023.

[35] M. de la Asunción, M. Castro, J. Mantas, and S. Ortega. Numerical simulation of tsunamis generated by landslides on multiple GPUs. *Advances in Engineering Software*, 99:5–2, 2016.

[36] P. De Mey-Frémaux, N. Ayoub, A. Barth, R. Brewin, G. Charria, F. Campuzano, S. Ciavatta, M. Cirano, C. A. Edwards, I. Federico, et al. Model-observations synergy in the coastal ocean. *Frontiers in Marine Science*, 6:436, 2019.

[37] D. P. Dee. Bias and data assimilation. *Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography*, 131(613):3323–3343, 2005.

[38] M. Destouches, P. Mycek, and S. Gürol. Multivariate extensions of the multilevel best linear unbiased estimator for ensemble-variational data assimilation. *arXiv preprint arXiv:2306.07017*, 2023.

[39] V. Desveaux and A. Masset. A fully well-balanced scheme for shallow water equations with Coriolis force. *Communications in Mathematical Sciences*, 20 (7):187–900, 2022.

[40] T. J. Dodwell, C. Ketelsen, R. Scheichl, and A. L. Teckentrup. A hierarchical multilevel Markov chain Monte Carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1075–1108, 2015. doi: 10.1137/130915005.

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

[42] D. Elfverson, F. Hellman, and A. Målqvist. A multilevel monte carlo method for computing failure probabilities. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):312–330, 2016.

[43] K. L. Elmore. Alternatives to the chi-square test for evaluating rank histograms from ensemble forecasts. *Weather and forecasting*, 20(5):789–795, 2005.

[44] G. Evensen. Sequential data assimilation with a nonlinear quasi-geostrophic model using monte carlo methods to forecast error statistics. *Journal of Geophysical Research: Oceans*, 99(C5):10143–10162, 1994.

[45] G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53:343–367, 2003.

[46] G. Evensen, F. C. Vossepoel, and P. J. van Leeuwen. *Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem*. Springer Nature, 2022.

[47] G. Evensen et al. *Data assimilation: The ensemble Kalman filter*, volume 2. Springer, 2009.

[48] C. Farmer. Upscaling: a review. *International journal for numerical methods in fluids*, 40(1-2):63–78, 2002.

[49] M. Fisher. Background error covariance modelling. In *Seminar on Recent Development in Data Assimilation for Atmosphere and Ocean*, pages 45–63. Shinfield Park, Reading, 2003.

[50] K. Fossum, T. Mannseth, and A. Stordal. Assessment of multilevel ensemble-based data assimilation for reservoir history matching. *Computational Geosciences*, 24(1):21–39, 2020. doi: 10.1007/s10596-019-09911-x.

[51] J. Galewsky, R. K. Scott, and L. M. Polvani. An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):429–440, 2004.

[52] M. B. Giles. Multilevel Monte Carlo path simulation. *Operations Research*, 56 (3):607–617, 2008. doi: 10.1287/opre.1070.0496.

[53] M. B. Giles. Multilevel Monte Carlo methods. *Acta numerica*, 24:259–328, 2015.

[54] M. B. Giles, T. Nagapetyan, and K. Ritter. Multilevel monte carlo approximation of distribution functions and densities. *SIAM/ASA journal on Uncertainty Quantification*, 3(1):267–295, 2015.

[55] T. Gneiting and M. Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1:125–151, 2014.

[56] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[57] T. Gneiting, F. Balabdaoui, and A. E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2):243–268, 2007.

[58] T. Gneiting, L. I. Stanberry, E. P. Grimit, L. Held, and N. A. Johnson. Assessing probabilistic forecasts of multivariate quantities, with an application to ensemble predictions of surface winds. *Test*, 17:211–235, 2008.

[59] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings F Radar and Signal Processing*, 140(2):107, 1993. doi: 10.1049/ip-f-2.1993.0015.

[60] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comput.*, 67(221):73–85, 1998. doi: 10.1090/s0025-5718-98-00913-2.

[61] A. Gregory. *Multilevel ensemble data assimilation*. PhD thesis, Imperial College London, 2017.

[62] A. Gregory and C. Cotter. On the calibration of multilevel monte carlo ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 143 (705):1929–1935, 2017.

[63] S. Griffies. *Fundamentals of ocean climate models*. Princeton university press, 2018.

[64] A. Gruber, M. Gunzburger, L. Ju, R. Lan, and Z. Wang. Multifidelity Monte Carlo estimation for efficient uncertainty quantification in climate-related modeling. *Geoscientific Model Development*, 16(4):1213–1229, 2023.

[65] K. Gundersen, A. Oleynik, N. Blaser, and G. Alendal. Semi-conditional variational auto-encoder for flow reconstruction and uncertainty quantification from limited observations. *Physics of Fluids*, 33(1), 2021.

[66] T. R. Hagen, M. O. Henriksen, J. M. Hjelmervik, and K.-A. Lie. How to solve systems of conservation laws numerically using the graphics processor as a high-performance computational engine. *Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF*, pages 211–264, 2007. doi: 10.1007/978-3-540-68783-2_8.

[67] T. Haiden, M. Dahoui, B. Ingleby, P. De Rosnay, C. Prates, E. Kuscu, T. Hewson, L. Isaksen, D. Richardson, H. Zuo, et al. Use of in situ surface observations at ECMWF. Technical report, European Centre for Medium Range Weather Forecasts, 2018.

[68] D. Ø. Halvorsen, A. Dallolio, and M. O. Alver. Assimilation of heterogeneous measurements at different spatial scales in the Arctic ocean and Norwegian sea. In *OCEANS 2022, Hampton Roads*, pages 1–10. IEEE, 2022. doi: 10.1109/oceans47191.2022.9977376.

[69] T. M. Hamill. Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560, 2001.

[70] S. Heinrich. Multilevel Monte Carlo methods. In *Large-Scale Scientific Computing: Third International Conference, LSSC 2001 Sozopol, Bulgaria, June 6–10, 2001 Revised Papers 3*, pages 58–67. Springer, 2001.

[71] H. Hoel, K. J. H. Law, and R. Tempone. Multilevel ensemble Kalman filtering. *SIAM Journal on Numerical Analysis*, 54(3):1813–1839, 2016. doi: 10.1137/15m100955x.

[72] H. H. Holm. *Efficient Forecasting of Drift Trajectories using Simplified Ocean Models and Nonlinear Data Assimilation on GPUs*. PhD thesis, NTNU, 2020.

[73] H. H. Holm, A. R. Brodtkorb, G. Broström, K. H. Christensen, and M. L. Sætra. Evaluation of selected finite-difference and finite-volume approaches to rotational shallow-water flow. *Communications in Computational Physics*, 27(4):1234–1274, 2020.

[74] H. H. Holm, A. R. Brodtkorb, and M. L. Sætra. GPU computing with Python: Performance, energy efficiency and usability. *Computation*, 8(1): 4, 2020. doi: 10.3390/computation8010004.

[75] H. H. Holm, M. L. Sætra, and A. R. Brodtkorb. Data assimilation for ocean drift trajectories using massive ensembles and GPUs. In *Finite Volumes for Complex Applications IX-Methods, Theoretical Aspects, Examples: FVCA 9, Bergen, Norway, June 2020 IX*, pages 715–723. Springer, 2020.

[76] H. H. Holm, M. L. Sætra, and P. J. Van Leeuwen. Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting. *Journal of Computational Physics: X*, 6:100053, 2020.

[77] P. L. Houtekamer and H. L. Mitchell. Data assimilation using an ensemble Kalman filter technique. *Monthly Weather Review*, 126(3):796–811, 1998.

[78] P. L. Houtekamer and H. L. Mitchell. A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137, 2001.

[79] P. L. Houtekamer and F. Zhang. Review of the ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 144(12):4489–4532, 2016.

[80] B. R. Hunt, E. J. Kostelich, and I. Szunyogh. Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126, 2007.

[81] M. Idžanović, E. S. U. Rikardsen, and J. Röhrs. Forecast uncertainty and ensemble spread in surface currents from a regional ocean model. *Frontiers in Marine Science*, 10, 2023. doi: 10.3389/fmars.2023.1177337.

[82] A. Jasra, K. Kamatani, K. J. H. Law, and Y. Zhou. Multilevel particle filters. *SIAM Journal on Numerical Analysis*, 55(6):3068–3096, 2017. doi: 10.1137/17m1111553.

[83] R. Keisler. Forecasting global weather with graph neural networks. *arXiv preprint arXiv:2202.07575*, 2022.

[84] Ø. Klemetsdal, O. Møyner, A. Moncorgé, H. Nilsen, K.-A. Lie, et al. High-resolution compositional reservoir simulation with dynamic coarsening and local timestepping for unstructured grids. *SPE Journal*, 26(06):4157–4173, 2021.

[85] N. M. Kristensen, L. P. Røed, and Ø. Sætra. A forecasting and warning system of storm surge events along the Norwegian coast. *Environmental Fluid Mechanics*, pages 1–23, 2022. doi: 10.1007/s10652-022-09871-4.

[86] S. Krumscheid and F. Nobile. Multilevel monte carlo approximation of functions. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1256–1293, 2018.

[87] A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Commun Math Sci*, 5(1):133–160, 2007. doi: 10.4310/cms.2007.v5.n1.a6.

[88] A. Kurganov, S. Noelle, and G. Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM J Sci Comput*, 23(3):707–740, 2001. doi: 10.1137/s1064827500373413.

[89] R. Lam, A. Sanchez-Gonzalez, M. Willson, P. Wirnsberger, M. Fortunato, F. Alet, S. Ravuri, T. Ewalds, Z. Eaton-Rosen, W. Hu, et al. GraphCast: Learning skillful medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.

[90] J. Latz, I. Papaioannou, and E. Ullmann. Multilevel sequential2 Monte Carlo for Bayesian inverse problems. *Journal of Computational Physics*, 368:154–178, 2018.

[91] P. F. Lermusiaux. Estimation and study of mesoscale variability in the Strait of Sicily. *Dynamics of Atmospheres and Oceans*, 29(2-4):255–303, 1999.

[92] R. LeVeque. *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press, 2004.

[93] R. J. LeVeque, I. M. Mitchell, and V. Stodden. Reproducible research for scientific computing: Tools and strategies for changing the culture. *Computing in Science & Engineering*, 14(04):13–17, 2012.

[94] K.-A. Lie. On Holden's seven guidelines for scientific computing and development of open-source community software. *preprint*, 2017.

[95] K.-A. Lie and S. Noelle. On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservation laws. *SIAM J Sci Comput*, 24(4):1157–1174, 2003. doi: 10.1137/s1064827501392880.

[96] K. Lye. *Computation of statistical solutions of hyperbolic systems of conservation laws*. PhD thesis, ETH Zurich, 2020.

[97] G. Madec and the NEMO team. *NEMO ocean engine*. Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27, 2008.

[98] M. J. Martin, M. Balmaseda, L. Bertino, P. Brasseur, G. Brassington, J. Cummings, Y. Fujii, D. Lea, J.-M. Lellouche, K. Mogensen, et al. Status and future of data assimilation in operational oceanography. *Journal of Operational Oceanography*, 8(sup1):s28–s48, 2015.

[99] A. Melsom, F. Counillon, J. H. LaCasce, and L. Bertino. Forecasting search areas using ensemble ocean circulation modeling. *Ocean Dynamics*, 62:1245–1257, 2012.

[100] S. Mishra and C. Schwab. Sparse tensor multi-level Monte Carlo finite volume methods for hyperbolic conservation laws with random initial data. *Mathematics of Computation*, 81(280):197–018, 2012. doi: 10.1090/s0025-5718-2012-02574-9.

[101] S. Mishra, C. Schwab, and J. Šukys. Multilevel Monte Carlo finite volume methods for shallow water equations with uncertain topography in multi-dimensions. *SIAM Journal on Scientific Computing*, 34(6):B761–B784, Jan. 2012.

[102] S. Mishra, U. Fjordholm, and R. Abgrall. Numerical methods for conservation laws and related equations. Lecture notes for Numerical Methods for Partial Differential Equations, ETH Zurich, 2019.

[103] M. Morzfeld, D. Hodyss, and C. Snyder. What the collapse of the ensemble Kalman filter tells us about particle filters. *Tellus A: Dynamic Meteorology and Oceanography*, 69(1):1283809, 2017.

[104] F. Müller. *Stochastic methods for uncertainty quantification in subsurface flow and transport problems*. PhD thesis, ETH Zurich, 2014.

[105] P. Mycek and M. De Lozzo. Multilevel Monte Carlo covariance estimation for the computation of Sobol' indices. *SIAM/ASA Journal on Uncertainty Quantification*, 7(4):132–348, 2019.

[106] M. Nezhadali. *Multilevel assimilation of inverted seismic data*. PhD thesis, University of Bergen, 2023.

[107] P. R. Oke, G. Larnicol, E. M. Jones, V. Kourafalou, A. Sperrevik, F. Carse, C. A. Tanajura, B. Mourre, M. Tonani, G. Brassington, et al. Assessing the impact of observations on ocean forecasts and reanalyses: Part 2, Regional applications. *Journal of Operational Oceanography*, 8(sup1):s63–s79, 2015.

[108] E. Ott, B. R. Hunt, I. Szunyogh, A. V. Zimin, E. J. Kostelich, M. Corazza, E. Kalnay, D. Patil, and J. A. Yorke. A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):415–428, 2004.

[109] J. Pathak, S. Subramanian, P. Harrington, S. Raja, A. Chattopadhyay, M. Mardani, T. Kurth, D. Hall, Z. Li, K. Azizzadenesheli, et al. FourCastNet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.

[110] J. Poterjoy. A localized particle filter for high-dimensional nonlinear systems. *Monthly Weather Review*, 144(1):59–76, 2016.

[111] J. Rabault, T. Nose, G. Hope, M. Müller, Ø. Breivik, J. Voermans, L. R. Hole, P. Bohlinger, T. Waseda, T. Kodaira, et al. OpenMetBuoy-v2021: An easy-to-build, affordable, customizable, open-source instrument for oceanographic measurements of drift and waves in sea ice and the open ocean. *Geosciences*, 12(3):110, 2022.

[112] A. E. Raftery, T. Gneiting, F. Balabdaoui, and M. Polakowski. Using bayesian model averaging to calibrate forecast ensembles. *Monthly weather review*, 133 (5):1155–1174, 2005.

[113] L. Røed. *Atmospheres and Oceans on Computers*. Springer, 2019.

[114] L. P. Røed. Documentation of simple ocean models for use in ensemble predictions. Technical Report 3, MET Norway, 2012.

[115] J. Röhrs, Y. Gusdal, E. Rikardsen, M. Durán Moro, J. Brændshøi, N. M. Kristensen, S. Fritzner, K. Wang, A. K. Sperrevik, M. Idžanović, et al. Barents-2.5 km v2.0: An operational data-assimilative coupled ocean and sea ice ensemble prediction model for the Barents Sea and Svalbard. *Geoscientific Model Development Discussions*, 2023:1–31, 2023.

[116] M. Sætra, A. Brodtkorb, and K.-A. Lie. Efficient GPU-implementation of adaptive mesh refinement for the shallow-water equations. *Journal of Scientific Computing*, 63:23–48, 2015.

[117] Ø. Saetra, H. Hersbach, J.-R. Bidlot, and D. S. Richardson. Effects of observation errors on the statistics for ensemble spread and reliability. *Monthly Weather Review*, 132(6):1487–1501, 2004.

[118] P. Sakov and L. Bertino. Relation between two common localisation methods for the EnKF. *Computational Geosciences*, 15:225–237, 2011.

[119] P. Sakov, F. Counillon, L. Bertino, K. Lisæter, P. Oke, and A. Korablev. TOPAZ4: an ocean-sea ice data assimilation system for the North Atlantic and Arctic. *Ocean Science*, 8(4):633–656, 2012. doi: 10.5194/os-8-633-2012.

[120] I. Samdal and B. Edvardsen. Massive salmon mortalities during a chrysochromulina leadbeateri bloom in northern norway. *Harmful Algae News*, 64:4–5, 2020.

[121] D. Schaden and E. Ullmann. On multilevel best linear unbiased estimators. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):601–635, 2020.

[122] D. W. Scott. *Multivariate density estimation: Theory, practice, and visualization*. John Wiley & Sons, 2015.

[123] A. F. Shchepetkin and J. C. McWilliams. The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9(4):347–404, 2005. doi: 10.1016/j.ocemod.2004.08.002.

[124] J. Skauvold, J. Eidsvik, P. J. Van Leeuwen, and J. Amezcua. A revised implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 145(721):1490–1502, 2019.

[125] C. Snyder, T. Bengtsson, P. Bickel, and J. Anderson. Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640, 2008.

[126] A. K. Sperrevik, K. H. Christensen, and J. Röhrs. Constraining energetic slope currents through assimilation of high-frequency radar observations. *Ocean Science*, 11(2):237–249, 2015.

[127] M. Sæatra. *Shallow Water Simulations on Graphics Hardware*. PhD thesis, University of Oslo, 2014.

[128] O. Talagrand, R. Vautard, and B. Strauss. Evaluation of probabilistic prediction systems. *Workshop on Predictability, 20-22 October 1997*, pages 1–26, 1997 1997.

[129] S. Taverniers, S. B. Bosma, and D. M. Tartakovsky. Accelerated multilevel monte carlo with kernel-based smoothing and latinized stratification. *Water Resources Research*, 56(9):e2019WR026984, 2020.

[130] P. G. Thoppil, S. Frolov, C. D. Rowley, C. A. Reynolds, G. A. Jacobs, E. Joseph Metzger, P. J. Hogan, N. Barton, A. J. Wallcraft, O. M. Smedstad, et al. Ensemble forecasting greatly expands the prediction horizon for ocean mesoscale variability. *Communications Earth & Environment*, 2(1):89, 2021.

[131] T. L. Thorarinsdottir, M. Scheuerer, and C. Heinz. Assessing the calibration of high-dimensional ensemble forecasts using rank histograms. *Journal of computational and graphical statistics*, 25(1):105–122, 2016.

[132] M. Tonani, M. Balmaseda, L. Bertino, E. Blockley, G. Brassington, F. Davidson, Y. Drillet, P. Hogan, T. Kuragano, T. Lee, et al. Status and future of global and regional ocean prediction systems. *Journal of Operational Oceanography*, 8(sup2):s201–s220, 2015.

[133] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[134] G. K. Vallis. *Essentials of atmospheric and oceanic dynamics*. Cambridge university press, 2019.

[135] P. J. Van Leeuwen. Efficient nonlinear data-assimilation in geophysical fluid dynamics. *Computers & Fluids*, 46(1):52–58, 2011.

[136] P. J. Van Leeuwen. A consistent interpretation of the stochastic version of the ensemble Kalman filter. *Quarterly Journal of the Royal Meteorological Society*, 146(731):2815–2825, 2020.

[137] P. J. Van Leeuwen, H. R. Künsch, L. Nerger, R. Potthast, and S. Reich. Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723):2335–2365, 2019.

[138] M. Verlaan, T. Zijlker, F. Zijl, and J. Sumihar. A bias-Kalman filter for operational storm-surge forecasting. In *9th International Symposium on Data Assimilation*, 2023.

[139] S. Vetra-Carvalho, P. J. Van Leeuwen, L. Nerger, A. Barth, M. U. Altaf, P. Brasseur, P. Kirchgessner, and J.-M. Beckers. State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems. *Tellus A: Dynamic Meteorology and Oceanography*, 70(1):1–43, 2018.

[140] P. A. Watson. Applying machine learning to improve simulations of a chaotic dynamical system using empirical error correction. *Journal of Advances in Modeling Earth Systems*, 11(5):1402–1417, 2019.

[141] J. S. Whitaker and T. M. Hamill. Ensemble data assimilation without perturbed observations. *Monthly weather review*, 130(7):1913–1924, 2002.

[142] D. Wilks. On the reliability of the rank histogram. *Monthly Weather Review*, 139(1):311–316, 2011.

[143] D. S. Wilks. *Statistical methods in the atmospheric sciences*, volume 100. Academic press, 2011.

[144] Y. Xing, C.-W. Shu, and S. Noelle. On the advantage of well-balanced schemes for moving-water equilibria of the shallow water equations. *Journal of scientific computing*, 48(–):339–349, 2011.

[145] M. Zhu, P. J. Van Leeuwen, and J. Amezcua. Implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 142(698):1904–1919, 2016.

Part II

Scientific Papers

Paper I

Reducing Numerical Artifacts by Sacrificing Well-Balance for Rotating Shallow-Water Flow

Håvard Heitlo Holm, Florian Beiser

Finite Volumes for Complex Applications X—Volume 2, Hyperbolic and Related Problems

DOI: 10.1007/978-3-031-40860-1_19

Reducing Numerical Artifacts by Sacrificing Well-Balance for Rotating Shallow-Water Flow

Håvard Heitlo Holm[1] and Florian Beiser[1,2]

[1]Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, NTNU, Trondheim, Norway

Abstract

We consider the problem of rotational shallow-water flow for which non-trivial rotating steady-state solutions are of great importance. In particular, we investigate a high-resolution central-upwind scheme that is well-balanced for a subset of these stationary solutions and show that the well-balanced design is the source of numerical artifacts when applied to more general problems. We propose an alternative flux evaluation that sacrifices the well-balanced property and demonstrate that this gives qualitatively better results for relevant test cases and real-world oceanographic simulations.

Keywords: high-resolution finite-volume schemes, rotating shallow-water equations, oceanography, numerical artifacts

1 Introduction

Over the past few years, we have developed a GPU-accelerated simulation framework [1] for oceanographic applications and ensemble prediction systems. At the core of the framework is a simplified ocean model based on the shallow-water equations in a rotational frame of reference. This hyperbolic system is solved using a slightly modified version of the numerical scheme proposed by Chertock et al. [2] (abbreviated as CDKLM). This is an explicit high-resolution finite-volume scheme that is conservative, second order, and consistent, formulated on Cartesian grids. Its main advantage compared to similar schemes is that it is well-balanced with respect to some non-trivial steady-state solutions in geostrophic balance, namely geostrophic jets along the coordinate axes. Hereby, the rotational pull on the jets is equal to the gravity-induced potential energy from a non-flat ocean surface. The geostrophic jets are however not the only steady-state solutions based on geostrophic balance, and in real-world oceanographic applications, rotational steady-states are often of greater importance.

The CDKLM scheme is based on central-upwind schemes [3], which are attractive as "black-box" solvers as they do not require solutions to the Riemann problem. An alternative to CDKLM is the earlier scheme proposed by Kurganov and Petrova [4] (abbreviated as KP), which also uses central-upwind flux evaluations but is not well-balanced for any non-trivial rotational steady-state solutions. When considering alternative numerical schemes, one that is well-balanced for some geostrophic steady-state solutions and one that is not, it is easy to choose to implement the former also for more complex applications. However, as we will show in this paper, one should be more careful in this selection.

In this work, we demonstrate that the CDKLM scheme gives oscillatory artifacts and show that we can achieve solutions that are qualitatively better for rotational flow by using a non-well-balanced variation of CDKLM. Thereafter, we continue by investigating how to tune the resulting finite-volume scheme when applied to real-world oceanographic problems.

2 Rotational Shallow Water Equations

The shallow water equations in a rotational frame of reference are given by

$$\begin{bmatrix} \eta \\ hu \\ hv \end{bmatrix}_t + \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}_x + \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ fhv \\ -fhu \end{bmatrix} + \begin{bmatrix} 0 \\ ghH_x \\ ghH_y \end{bmatrix} \tag{1}$$

and conserve mass and momentum. The variable η represents the water surface as a deviation from the mean equilibrium depth H, so that $h = H + \eta$ becomes the total water depth. Momentum is given by hu and hv in x- and y-direction, respectively. Furthermore, g is the gravitational constant, and f is the Coriolis parameter, which is a function of latitude. We define the vector $\mathbf{q} = [\eta, hu, hv]^T$ to contain the conserved variables, so that we can write (1) as

$$\mathbf{q}_t + \boldsymbol{\mathcal{F}}(\mathbf{q})_x + \boldsymbol{\mathcal{G}}(\mathbf{q})_y = \mathbf{S}(\mathbf{q}) \, . \tag{2}$$

Rotating steady-states solutions of (1) are given by geostrophic balances, where

$$u_x + v_y = 0 \, , \qquad g\eta_x = fv \, , \quad \text{and} \quad g\eta_y = -fu \, . \tag{3}$$

In [2], the last two expressions are rewritten in terms of potential energies as

$$K(x,y) := g\left(\eta - \int_{-\infty}^{x} \frac{f}{g} v \, \mathrm{d}x'\right), \qquad L(x,y) := g\left(\eta - \int_{-\infty}^{y} \frac{f}{g} u \, \mathrm{d}y'\right), \tag{4}$$

so that they can be written as $K_x = 0$ and $L_y = 0$. In particular, the CDKLM scheme is well-balanced with respect to the geostrophic jets

$$u \equiv 0, \quad v_y \equiv 0, \quad \eta_y \equiv 0, \quad H_y \equiv 0, \quad K \equiv \text{const}, \tag{5}$$

$$v \equiv 0, \quad u_x \equiv 0, \quad \eta_x \equiv 0, \quad H_x \equiv 0, \quad L \equiv \text{const}, \tag{6}$$

which are special cases of (3).

3 Finite Volume Schemes

In the following, we summarize only the relevant parts of the CDKLM scheme; the reader can find all details in [2]. We start by discretizing the domain in a regular Cartesian grid with cells of size $(\Delta x, \Delta y)$, and define $\mathbf{Q}_{j,k}$ to be the average of $\mathbf{q}$ within the cell with center at $\big((j+\frac{1}{2})\Delta x, \, (k+\frac{1}{2})\Delta y\big)$. Equation (2) can then be written in semi-discrete form as

$$\frac{\partial \mathbf{Q}_{j,k}}{\partial t} = -\frac{\mathbf{F}_{j+1/2,k} - \mathbf{F}_{j-1/2,k}}{\Delta x} - \frac{\mathbf{G}_{j,k+1/2} - \mathbf{G}_{j,k-1/2}}{\Delta y} + \mathbf{S}(\mathbf{Q}_{j,k}). \tag{7}$$

The system is solved in time with the second-order total-variation-diminishing Runge-Kutta scheme [5] with time steps satisfying the CFL condition. In the rest of the paper, we focus on the spatial reconstruction and fluxes.

3.1 Well-balanced scheme

To design a well-balanced finite-volume scheme for (1) with respect to (5) and (6), Chertock et al. [2] propose to reconstruct the face values of $\mathbf{q}$ by computing the slopes of the equilibrium variables $\mathbf{p} = [u, v, K, L]^T$ rather than the slopes of $\mathbf{q}$ directly. Values for $K_{j,k}$ and $L_{j,k}$ are obtained by evaluating the integrals in (4) by a recursive sum starting from zero at the boundary. Note that the recursive sums for K and L mostly

cancel in the reconstruction and are reduced to local operations [1, Sect. 3.1]. The slopes are then found by the generalized minmod limiter (c.f. [6]), which in the x-direction is

$$(\mathbf{p}_x)_{j,k} = \text{minmod}\left(\theta\frac{\mathbf{p}_{j+1,k}-\mathbf{p}_{j,k}}{\Delta x}, \frac{\mathbf{p}_{j+1,k}-\mathbf{p}_{j-1,k}}{2\Delta x}, \theta\frac{\mathbf{p}_{j,k}-\mathbf{p}_{j-1,k}}{\Delta x}\right), \tag{8}$$

where $\theta \in [1,2]$ controls numerical dissipation.

Using $(\mathbf{p}_x)_{j,k}$ and $(\mathbf{p}_x)_{j+1,k}$, we can reconstruct one-sided point values for $\mathbf{q}$ at the face between the two cells as $\mathbf{Q}^r_{j+1/2,k}$ and $\mathbf{Q}^l_{j+1/2,k}$ from the right and left, respectively, using the discretized version of (4) to reconstruct values for η. The flux terms in (7) are computed using the central-upwind (CU) scheme [3]

$$\mathbf{F}^{(i)}_{j+1/2,k} = \frac{a^+\boldsymbol{\mathcal{F}}\left(\mathbf{Q}^l\right) - a^-\boldsymbol{\mathcal{F}}\left(\mathbf{Q}^r\right)}{a^+ - a^-} + \frac{a^+a^-}{a^+-a^-}\left(\mathbf{Q}^r - \mathbf{Q}^l\right) \tag{9}$$

for $i = 1, 2$, where we have omitted the subscript $(\cdot)_{j+1/2,k}$ for all $\mathbf{Q}$ (including its entries) and a variables for readability. The variables a^+ and a^- represent the largest positive and negative wave speeds at the cell interface, respectively,

$$a^\pm = \begin{matrix}\max\\ \min\end{matrix}\left\{u^l \pm \sqrt{gh^l}, \quad u^r \pm \sqrt{gh^r}, \quad 0\right\}. \tag{10}$$

For the scheme to be well-balanced with respect to (6), however, Chertock et al. [2] propose to use a standard-upwind (SU) scheme for $\mathbf{F}^{(3)}_{j+1/2,k}$, given by

$$\mathbf{F}^{(3)}_{j+1/2,k} = \begin{cases} h^r u^r v^r, & \text{if } u^r + u^l > 0,\\ h^l u^l v^l, & \text{otherwise.}\end{cases} \tag{11}$$

The scheme is defined analogously in the y-direction, where the CU flux in (9) is used for $\mathbf{G}^{(i)}_{j,k+1/2}$, $i = 1, 3$, and the SU in (11) is used for $\mathbf{G}^{(2)}_{j,k+1/2}$.

3.2 Unbalanced alternatives

As we will see in Section 4, the use of the SU flux in (11) produces artifacts when applying CDKLM to rotational and realistic problems. To investigate the problem, we consider a convex combination of the two fluxes

$$\mathbf{F}^{(3)}_{j+1/2,k} = \phi(\mathbf{F}_{SU})^{(3)}_{j+1/2,k} + (1-\phi)(\mathbf{F}_{CU})^{(3)}_{j+1/2,k}, \tag{12}$$

in which $\mathbf{F}_{CU}$ comes from (9), $\mathbf{F}_{SU}$ comes from (11), and $\phi \in [0, 1]$.

In contrast to CDKLM, KP reconstructs the slopes of $\mathbf{q}$ using the conserved variables directly and evaluates all flux components using the central-upwind scheme. We implement (12) also for KP and compare it with CDKLM to better assess the effects of different reconstructions and flux terms in isolation.

4 Numerical Experiments

We consider five experiments to evaluate the pros and cons of using the reconstruction and flux that make the scheme well-balanced for geostrophic jets.

Case A: Radial wave without rotation. First, we consider a simple radial wave in a non-rotating domain (meaning $f = 0$), initialized by

$$\eta(x, y, 0) = \exp^{-10^{-5}(x^2+y^2)}, \quad \text{and} \quad hu(x, y, 0) = hv(x, y, 0) = 0. \tag{13}$$

The domain consists of 400×400 cells with $\Delta x = \Delta y = 100\,\text{m}$, centered at the origin. The equilibrium depth is $H(x, y) = 60\,\text{m}$.

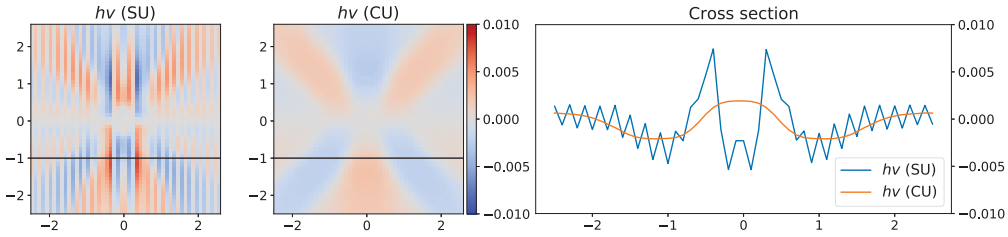


Figure 1: Artifacts in hv from an initial bump using CDKLM with SU flux (left) and CU flux (center), and cross section of hv at $y = -1\,\mathrm{km}$ (right).

Figure 1 shows hv at $t = 1000\,\mathrm{s}$ in the central part of the domain after most of the energy has traveled outwards. In the plots of the y-momentum shown in the two leftmost plots for the CDKLM scheme with SU and CU fluxes, we clearly see an oscillating artifact when using SU. This is even more pronounced in the right-hand plot of Fig. 1, which shows a cross section at $y = -1\,\mathrm{km}$. The CU flux gives numerical artifacts along the grid diagonals, but these are smooth in contrast to the SU artifacts. Both solutions are symmetric in hv and hu, and the artifacts are observed for any spatiotemporal resolution satisfying the CFL conditions with Courant number less or equal to 0.5.

Note that with $f = 0$, the CDKLM reconstruction reduces to that of the the KP scheme. We have implemented the KP scheme independently, and get exactly the same results as for CDKLM using both types of fluxes. This excludes the possibility that the artifacts are caused by a programming bug.

Case B: Rossby adjustment. We continue by considering Rossby adjustment, in which a rotational steady state is generated when running Case A with $f = 0.0012\,\mathrm{s}^{-1}$. The presence of Coriolis forces will establish a rotating bump in the center of the domain and create a steady-state that is not covered by the jets described in (5) and (6).

In Fig. 2, we consider the flux in (12) and assess different values for ϕ. The leftmost plot uses $\phi = 1.0$, which corresponds to a pure SU flux as in the original CDKLM scheme, whereas $\phi = 0.0$ to the far right corresponds to the CU flux. Again, hu and hv are symmetric, so we only show results for hv. We see a similar nonphysical oscillation pattern as in Case A when using the SU flux, and the rotating bump is dragged along the grid axes. In contrast, the CU flux results in a circular bump that captures (3) much better. It is interesting that it is sufficient to introduce just a small CU contribution to the SU flux to improve the numerical behavior, and the results are already significantly better with $\phi = 0.9$. The figures are visually the same when using KP.

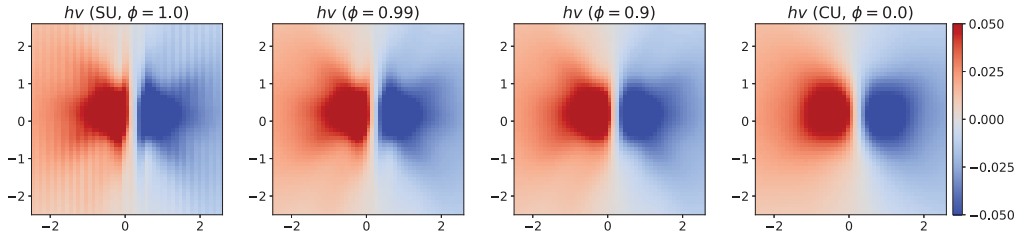


Figure 2: Numerical steady-states in hv from an initial bump with different ϕ-values in flux (12) in the presence of Coriolis forces.

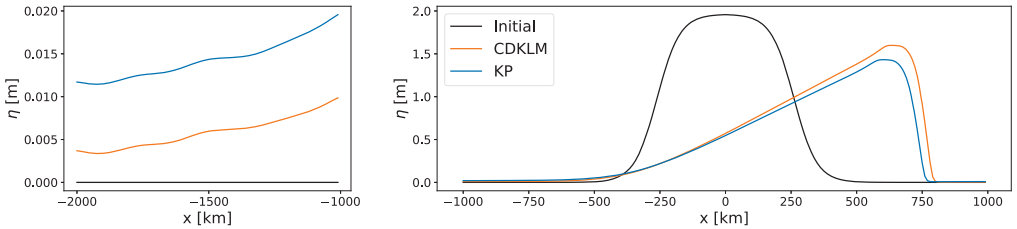


Figure 3: Cross-sectional plots of a Kelvin wave after ten periods for CDKLM and KP with CU flux and $\theta = 1.5$. Tail on the left and wave center on the right panel.

Case C: Kelvin wave. Kelvin waves are relatively fast travelling waves trapped along the coast by the Coriolis forces. They are travelling with the coast to their right on the Northern hemisphere and are important in realistic applications, as the tides often form them. Here, we consider a periodic domain in the east-west direction with wall boundary conditions in the north and south, and initialize a wave travelling eastwards along the southern wall. Sufficiently small waves follow linear physics and give periodic solutions, but the amplitudes used herein give nonlinear responses that develop a shock over time; see [7, Sect. 4.4] for details.

Figure 3 compares the CDKLM and KP schemes when using the CU flux and $\theta = 1.5$ for both, meaning that the only difference is in the slope reconstruction. The figure shows the solution after ten periods, corresponding to more than 60,000 time steps, at two cross sections for fixed y. We see how both schemes capture the nonlinear behavior of large waves. CDKLM manages, however, to maintain a taller wave front than KP, whereas KP produces a long nonzero tail. We argue that these results favor CDKLM for this case.

Case D: Geostrophic jets. We consider a double geostrophic jet that satisfies the steady-state given by (6), with one jet travelling eastward in the northern part of the domain and an westward jet in the southern part. We use periodic boundary conditions; the case is described in full in [8, Sect. 5]. This case exposes what we lose by not using the well-balanced SU flux for CDKLM, as this flux preserves this steady state for any slope limiter θ.

Figure 4 shows a cross section along y after running CDKLM for a long time using the CU flux with different values for θ. With $\theta = 1.0$, the numerical dispersion smooths out the jet streams significantly, but a moderate increase in θ compensates for using a non-balanced flux, and the jets stay captured and only cap the extrema slightly. With the maximal possible $\theta = 2.0$, we lose only 4.5% of the momentum in the center

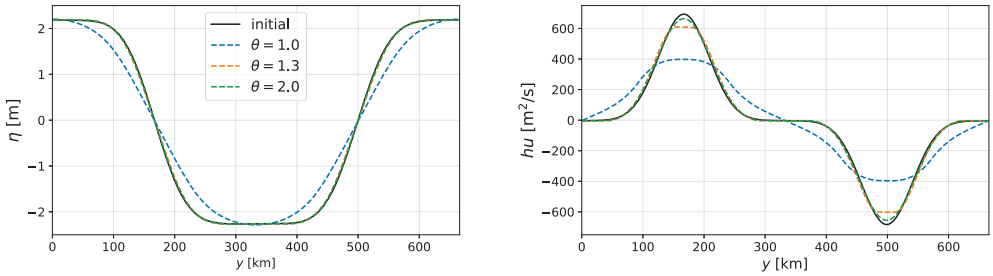


Figure 4: A double jet state after 9 simulation days with the CU flux: the plots show cross-sections of η (left) and hu (right).

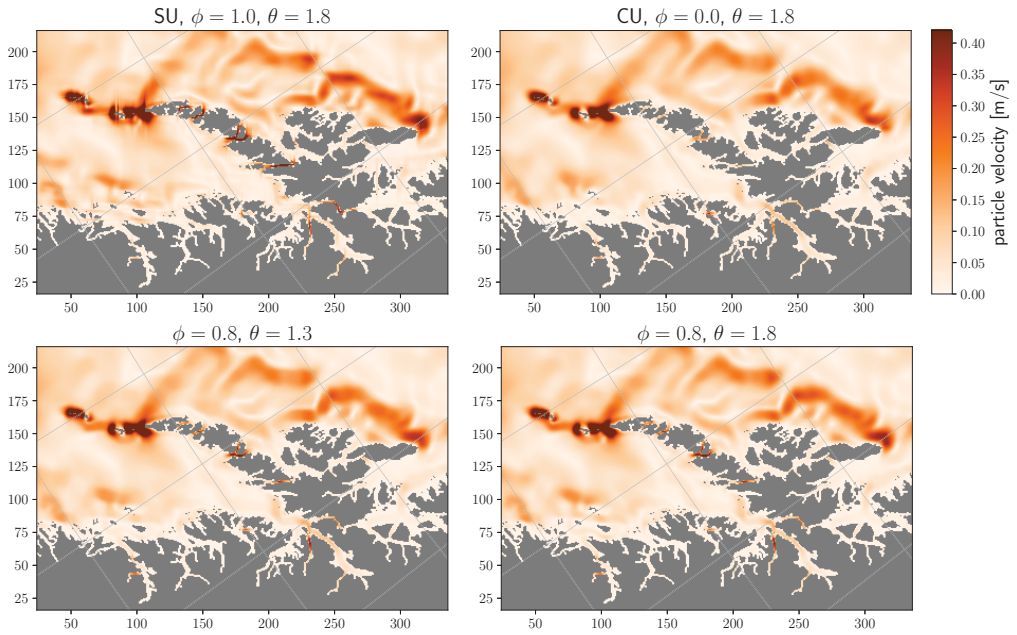


Figure 5: Real-world simulation for the Lofoten archipelago in Northern Norway. The axes show position within the computational domain in km.

of the jets.

To see how much of the CU flux we can incorporate without diminishing the preservation of geostrophic jets, we run the experiment for decreasing values of ϕ with fixed $\theta = 1.8$. For all $\phi \geq 0.52$, we conserve the maximum momentum within 4.5%.

Case E: Real-world simulation. Finally, we study the consequence of using different flux terms and slope limiters when using the CDKLM scheme for simulating a real-world oceanographic scenario. Here, we utilize the full GPU-accelerated framework from [1], in which land mask, bathymetry, initial and boundary conditions, domain orientation, and wind forcing are taken from the operational model NorKyst800 [9], with 800 m horizontal resolution.

Figure 5 shows water velocity at 23 h for a few selected values of ϕ and θ for CDKLM. The first thing to notice is how the SU flux gives visible artifacts around the outermost islands at $(100, 150)$. Secondly, the CU scheme gives visibly slower currents than SU, e.g., at $(300, 150)$. Since the tidal Kelvin wave follows the coast from left towards right, the Lofoten islands trap the tidal wave, causing strong currents in between the islands. With CU, we do not resolve this, but setting $\phi = 0.8$ helps significantly. It is hard to see visual differences between setting $\theta = 1.3$ and $\theta = 1.8$, but examining cross sections (not shown) reveals that we indeed get sharper fronts but no oscillations with the latter, which is preferred.

5 Conclusion

We have demonstrated numerical artifacts in the CDKLM scheme in cases that differ from geostrophic balanced jets along the coordinate axes. We identified the source of the oscillations to be the standard-

upwind flux in the *huv* fluxes. By combining the standard-upwind flux with the central-upwind flux with a factor ϕ, we reduced unintended numerical behavior significantly and made the results more physical, even though we lose the well-balanced property for the jets.

We have illustrated the artifacts in Cases A and B, where Case B indicates that combining the SU flux with a small factor of CU is sufficient to smooth out these oscillations. In Case C, we see that the slope reconstruction in CDKLM still gives a small advantage over the more standard reconstruction used by the KP scheme. By using a large value θ in the slope limiter, we demonstrated that also pure CU can preserve the geostrophic jets from Case D within acceptable tolerances. Case E, however, shows that the pure CU flux is unable to capture strong realistic currents in narrow straits. The conclusion is then to use values close to $\phi = 0.8$ and $\theta = 1.8$ as flux parameter and slope limiter, respectively, for real-world oceanographic applications. This reduces the observed artifacts and gives a practical compromise between the well-balanced SU flux and the smoother behaving CU flux.

Acknowledgement This work is supported by the Research Council of Norway (RCN) through grant number 310515 (Havvarsel). The numerical examples are made with the GPU Ocean software [10], and can be reproduced by [11].

References

[1] A. Brodtkorb and H. Holm. Coastal ocean forecasting on the GPU using a two-dimensional finite-volume scheme. *Tellus A*, 73(1):1–22, 2021.

[2] A. Chertock, M. Dudzinski, A. Kurganov, and M. Lukácová-Medvidová. Well-balanced schemes for the shallow water equations with Coriolis forces. *Numer. Math.*, 138(4):939–973, 2017.

[3] A. Kurganov, S. Noelle, and G. Petrova. Semidiscrete central-upwind schemes for hyperbolic conservation laws and Hamilton-Jacobi equations. *SIAM J Sci Comput*, 23(3):707–740, 2001.

[4] A. Kurganov and G. Petrova. A second-order well-balanced positivity preserving central-upwind scheme for the Saint-Venant system. *Commun Math Sci*, 5(1):133–160, 03 2007.

[5] S. Gottlieb and C.-W. Shu. Total variation diminishing Runge-Kutta schemes. *Math. Comput.*, 67(221):73–85, 1998.

[6] K.-A. Lie and S. Noelle. On the artificial compression method for second-order nonoscillatory central difference schemes for systems of conservation laws. *SIAM J Sci Comput*, 24(4):1157–1174, 2003.

[7] H. Holm, A. Brodtkorb, G. Broström, K. Christensen, and M. Sætra. Evaluation of selected finite-difference and finite-volume approaches to rotational shallow-water flow. *Commun Comput Phys*, 27(4):1234–1274, 2020.

[8] H. Holm, M. Sætra, and P. van Leeuwen. Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting. *J. Comput. Phys.: X*, 6(0314):100053, 2020.

[9] J. Albretsen, A. Sperrevik, A. Staalstrøm, A. Sandvik, and F. Vikebø. NorKyst-800 report no. 1: User manual and technical descriptions. Technical Report 2, Fisken og Havet, Institute of Marine Research, 2011.

[10] A. Brodtkorb, H. Holm, M. Sætra, and F. Beiser. GPU Ocean, 2023. DOI: 10.5281/zenodo.7938844.

[11] H. Holm and F. Beiser. Supplementary material for "Reducing numerical artifacts by sacrificing well-balance for rotating shallow-water flow", 2023. DOI: 10.5281/zenodo.7938887.

Paper II

Combining Barotropic and Baroclinic Simplified Models for Drift Trajectory Predictions

Florian Beiser, Håvard Heitlo Holm, Martin Lilleeng Sætra, Nils Melsom Kristensen, Kai Håkon Christensen

In review

Combining Barotropic and Baroclinic Simplified Models for Drift Trajectory Predictions

Florian Beiser[1,3], Håvard Heitlo Holm[1], Martin Lilleeng Sætra[2,4], Nils Melsom Kristensen[2], and Kai Håkon Christensen[2,5]

[1]Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[2]Norwegian Meteorological Institute, Oslo, Norway
[3]Department of Mathematical Sciences, NTNU, Trondheim, Norway
[4]Department of Computer Science, Oslo Metropolitan University, Oslo, Norway
[5]Department of Geosciences, University of Oslo, Oslo, Norway

Abstract

The shallow-water equations are often used as a classical simplified ocean model for barotropic ocean dynamics. The same equations can also be used to model simplified baroclinic dynamics through the reduced-gravity model. Herein, we propose to utilize a GPU-accelerated shallow-water simulation framework for representing two decoupled simplified ocean models for each of the barotropic and baroclinic dynamics, and use these models for ensemble prediction of short-term drift trajectories in coastal domains. This system can be used complementary to current operational systems as a lightweight tool for uncertainty quantification and in the future for ensemble-based data assimilation of in-situ observations. We show the relevance of our approach by demonstrating how the barotropic and baroclinic dynamical components are of varying importance at different geographical locations, and that one of the models can be used alone or in combination with the other. We benchmark our approach by showing the resulting ensemble trajectories in reference to deterministic trajectories produced by operational models for three Norwegian coastal regions.

Keywords: Ocean Modeling, Shallow-Water Equations, GPU Computing, Ensemble Prediction Systems, Drift Trajectories, Decision-Support

1 Introduction

The development of ocean forecasting systems is generally towards increasing spatial resolution. Operational forecasts are typically constrained by the availability of computational resources, meaning the configuration and size of the available compute cluster and the prioritization among other operational and non-operational compute-intensive jobs. When spatial resolution is of particular importance, for example as in regional models of complex circulation in the coastal zone, resources are often used for a single deterministic ocean forecast with the highest level of detail that fits within the available computational resources and still produces a timely forecast. The result is then a detailed deterministic forecast, but with a lack of uncertainty quantification.

Herein, we describe an ensemble prediction system based on two simplified ocean models that can be considered complementary to fully three-dimensional, primitive equation circulation models (Madec and the NEMO team, 2008; Shchepetkin and McWilliams, 2005; Bleck, 2002; Chassignet et al., 2003). In this context, "simplified" means that we do not attempt to produce the most realistic representation of the ocean dynamics, but rather develop lightweight models that complement the resource-demanding fully three-dimensional models. The aim is to provide short-term ensemble forecasts of the upper ocean circulation for

decision support (Christensen et al., 2018). Since we only seek some predictive ability on short timescales (typically a few hours), we do not seek to model any changes in hydrography due to air-sea interactions, river runoff, or turbulent mixing, but limit ourselves to the classical problem of wave propagation, solving a set of hyperbolic equations for (i) the depth-averaged current (barotropic mode), and (ii) the upper mixed layer (lowest order baroclinic mode). It should be noted that we decouple these modes by assuming a 1.5-layer/reduced-gravity model for the upper ocean. As discussed further below, a coupled two-layer approach is certainly feasible, but our interest here is in the most lightweight system that is still fit for the purpose, exploiting the benefits of having decoupled baroclinic and barotropic modes. Other elements of the upper ocean dynamics relevant to oceanic drift, i.e., the wind drag and wind- and wave-induced currents, are parameterized in an accompanying online trajectory model.

The objective of our approach is thus fast on-demand simulations in limited areas, potentially with complex topography; better representation of uncertainty employing large model ensembles; and, further down the line, enabling nonlinear assimilation of in-situ observations from, e.g., drifting buoys (Holm et al., 2020b,c). The framework we have developed runs almost exclusively on graphics processing units (GPUs), further speeding up simulations while reducing watts per flops (see, e.g., Huang et al. (2009); Dong et al. (2014)). Our interest is in drift trajectory forecasting, which is a crucial component in coastal preparedness and emergency management. Applications range from search-and-rescue operations, planning and organizing clean-up of spilled pollutants, to decision support for large drifting objects such as abandoned ships or icebergs. In this context, we are interested in predicting the ocean currents to forecast probabilistic trajectories, with a requirement to do so within tight time frames.

In our previous work, we have presented a GPU-accelerated simulation framework for running shallow-water ocean models (Røed, 2019) on realistic domains (Brodtkorb and Holm, 2021). We refer to this framework as "GPU Ocean". The motivation has been to explore heterogeneous computing for storm-surge modeling (Kristensen et al., 2022), and we have also been using it for probabilistic forecasting of drift trajectories using various methods for data assimilation (Holm et al., 2020b,c; Beiser et al., 2023). We have used high-resolution finite volume schemes, and a significant effort has been made to make these schemes well balanced for geophysical (rotational) flows and applicable for realistic topography, while optimizing the code for fast simulations. A weakness of this approach, however, is that we neglect the baroclinic dynamics driven by spatial differences in salinity and temperature, which is an important contribution to particle drift (Röhrs and Christensen, 2015).

In this work, we combine the barotropic simplified ocean model from Brodtkorb and Holm (2021) with a reduced-gravity model for the upper mixed layer of the ocean. We use the same GPU-accelerated computational framework for both models and initialize and force our simulations using available deterministic forecasts made from traditional coastal ocean forecasting system. Depending on the local conditions, in some cases only the contribution of only one of the simplified models may be relevant, and in other cases both models may be needed. Then, we compute drift trajectories assuming the drift is a sum of the baroclinic and barotropic currents. While perturbing various input parameters for the trajectory modeling, such as wind strength and direction, we facilitate the construction of large drift trajectory ensembles through the cross products between smaller barotropic and larger baroclinic ensembles.

Related Work. Reduced-gravity models were developed early on for modeling of the mixed layer dynamics, cf. Gent and Cane (1989); Arango and Reid (1991); Kim and Yoon (1996), and Dellar and Salmon (2005) also used the shallow-water equations to do so. Reduced-gravity models can also be useful for other applications, e.g., Samelson et al. (2019) use a one-layer, reduced-gravity, quasigeostrophic model to simulate open-ocean mesoscale variability; and Wang et al. (2003) employ a two-layer, reduced-gravity model to simulate dense water flow on continental shelves.

An alternative to our proposed combination of decoupled baroclinic and barotropic shallow-water models, is to use the coupled two-layer shallow-water equations as the simplified ocean model. The two-layer model assumes vertical stratification of the ocean, with two shallow-water layers interacting by the dynamic depth of their intersection (see, e.g. Røed (2012) for derivation), and accounts for baroclinic and barotropic physics. Although apparently well-suited for our application and proved to be suited for GPU-acceleration (de la

Asunción et al., 2016), numerical implementation of the two-layer model is more challenging due to non-conservative terms and possible loss of hyperbolicity (Castro-Díaz et al., 2011). Here, we reuse existing code for the barotropic simulator, hence a limited amount of work is needed to extend existing one-layer shallow-water codes. Having two separate models is also more flexible, making it possible to turn on and off modes depending on the case and domain. Furthermore, we are able to span an ensemble by combining different pairs of barotropic and baroclinic realizations.

In the work by Kudryavtsev et al. (2019), a simplified ocean model is built under the distinct consideration of barotropic and baroclinic modes for the application of tropical cyclones – for this special scenario, they find analytical solutions under the assumption of a non-changing mixed-layer depth. Peña-Molino et al. (2014) treat barotropic and baroclinic mass transport across the Antarctic Circumpolar current separately, as they are, on average, in opposite directions, and they find that barotropic and baroclinic transport is spatially variable.

Herein, we rely on an online built-in trajectory model (see Section 2.3), with the OpenDrift (Dagestad et al., 2018) trajectory framework used as a reference. The OpenDrift package is a part of the emergency preparedness at the Norwegian Meteorological Institute, and has internal models for oil spills, disabled ships drift forecasting, search and rescue operations, and iceberg monitoring. It uses 3D fields of the ocean state to make an offline trajectory forecast. For a thorough survey of drift in the ocean and trajectory modelling we refer the reader to Röhrs et al. (2023) and van Sebille et al. (2020), and references therein.

Outline. The article is structured as follows: Section 2 describes how the shallow-water equations are used for both the barotropic and the baroclinic parts of the simplified ocean model. Numerical results for idealized conditions to show the validity of the reduced-gravity model are presented in Section 3. Section 4 showcases the application of the purely barotropic, purely baroclinic, and combined barotropic-baroclinic models for drift trajectory forecasting in ensemble prediction systems (EPSs) for three different Norwegian coastal areas. The paper ends with a discussion in Section 5.

2 Simplified Ocean Models for Barotropic and Baroclinic Motion

In the following, we describe the barotropic and baroclinic versions of our simplified ocean model, and how we combine the currents from the two independent models for drift trajectory modeling. A complete derivation of the barotropic model can be found in Røed (2019), whereas the computational simulation framework used herein is presented in detail in Brodtkorb and Holm (2021). We will nevertheless give a brief overview of the barotropic model for completeness and to emphasize its similarities and differences from the baroclinic model. Since we will show that the two models can be represented by the same mathematics, we will use $\bar{\cdot}$ and $\tilde{\cdot}$ to represent variables belonging to the barotropic and the baroclinic models, respectively, when the same symbols are needed. To improve readability, we will also abuse this notation so that, e.g., $\overline{hu}$ represents $\bar{h}\bar{u}$.

2.1 Barotropic Model

The shallow-water equations are a well-established barotropic ocean model, which can be derived under simplifying assumptions from the primitive equations. The equations form a hyperbolic conservation law, conserving mass and momentum within a rotating frame of reference. They model depth-averaged horizontal velocities and thereby neglect the vertical component of the momentum. This results in a single-layer two-dimensional model, that can be written as

$$\begin{bmatrix} \eta \\ \overline{hu} \\ \overline{hv} \end{bmatrix}_t + \begin{bmatrix} \overline{hu} \\ \overline{hu^2} + \frac{1}{2}g\overline{h^2} \\ \overline{huv} \end{bmatrix}_x + \begin{bmatrix} \overline{hv} \\ \overline{huv} \\ \overline{hv^2} + \frac{1}{2}g\overline{h^2} \end{bmatrix}_y = \begin{bmatrix} 0 \\ f\overline{hv} \\ -f\overline{hu} \end{bmatrix} + \begin{bmatrix} 0 \\ g\overline{h}\overline{H_x} \\ g\overline{h}\overline{H_y} \end{bmatrix} + \begin{bmatrix} 0 \\ \overline{\tau}_u \\ \overline{\tau}_v \end{bmatrix} + \begin{bmatrix} 0 \\ -\overline{r}\overline{u}\sqrt{\overline{u}^2 + \overline{v}^2} \\ -\overline{r}\overline{v}\sqrt{\overline{u}^2 + \overline{v}^2} \end{bmatrix}. \quad (2.1)$$

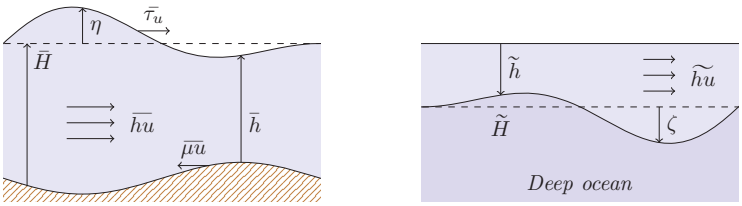


Figure 2.1: Overview of variables used in the shallow-water models, here shown in one dimension. To the left, we see the barotropic model covering the entire water column, and to the right we see the baroclinic model capturing the dynamics within the mixed layer.

Here, and as shown in Figure 2.1, η is the sea-surface deviation from mean equilibrium level and $\bar{H}$ is the static equilibrium depth, meaning that $\bar{h} = \bar{H} + \eta$ is the total water depth. The momentum is then denoted by $\overline{hu}$ and $\overline{hv}$ along the x- and y-axis, respectively, and subscripts t, x and y are used for temporal and spatial derivatives. Furthermore, g is the gravitational constant, f is the Coriolis parameter accounting for the rotating frame of reference, and $\bar{\tau} = [\bar{\tau}_u, \bar{\tau}_v]$ is the wind stress calculated as in Large and Pond (1981). The final source term accounts for bed friction and is modeled by a parameterized nonlinear function inspired by the Manning formula (see Dyakonova and Khoperskov (2018)), with dimensionless parameter $\bar{r}$.

To simulate (2.1), we use the GPU-accelerated simulation framework presented in Brodtkorb and Holm (2021). The framework is designed to run barotropic simulations on realistic domains and can be initialized by operational 3D ocean forecasts[1] and optionally forced by weather forecast parameters such as wind and atmospheric pressure. It can run with both higher and lower resolution than what has been used operationally, and supports nested simulations where the operational forecast is used as boundary conditions. The computational grid is assumed to be uniform with grid cells of size $(\Delta x, \Delta y)$, but can have arbitrary orientation, meaning that the Coriolis parameter f varies along the direction towards north, and not along the y-axis as is often the case.

Within the framework, we solve (2.1) numerically by a slightly modified version of the well-balanced high-resolution finite volume scheme proposed by Chertock et al. (2018). The scheme is designed to be well-balanced with respect to special cases of geostrophic balance, and is of second order. We solve the equations with an explicit second-order strong stability-preserving Runge-Kutta scheme (Gottlieb and Shu, 1998), with time step lengths restricted by the CFL-condition,

$$\Delta \bar{t} \leqslant \frac{1}{4} \min \left\{ \frac{\Delta x}{\max \left| \bar{u} \pm \sqrt{g\bar{h}} \right|}, \frac{\Delta y}{\max \left| \bar{v} \pm \sqrt{g\bar{h}} \right|} \right\}. \tag{2.2}$$

GPUs are designed based on the single-instruction multiple-data (SIMD) paradigm (NVIDIA, 2023; Sanders and Kandrot, 2010), in which a large set of computational cores carry out the same set of mathematical operations on different data. Explicit numerical schemes for solving partial differential equations, such as the one we use to solve (2.1), are particularly well-suited for implementation on such hardware, as these schemes can be said to be *embarrassingly parallel*. This means that each updated discrete variable can be found independently from all other updated variables, which allows for all of the variables to be updated completely in parallel. This is in contrast to implicit methods, where each updated variable also depends

[1]The simulation framework has mainly been developed using ocean forecasts from the NorKyst-800 model (Albretsen et al., 2011) that is operationally run by the Norwegian Meteorological Institute. Other forecast models have been tested as well, and in this study in particular, we use data from NorFjords-160 and FjordOs (Røed et al., 2016), the latter focuses on the Oslofjord.

on the values of other updated variables. For early examples of GPU-accelerated shallow-water simulations, see (Brodtkorb et al., 2012) and (de la Asunción et al., 2011).

2.2 Baroclinic Model

The ocean is often stratified with a well defined pycnocline. It separates the upper layer, which is typically a $O(10\,\mathrm{m} - 100\,\mathrm{m})$ thick layer of well-mixed water with density ρ_1, from the deep lower layer with density ρ_2. The upper mixed layer is directly influenced by winds and breaking waves, while the ocean interior is more weakly coupled to the atmospheric forcing. The mixed-layer depth (MLD) is neither static nor spatially uniform, and the spatial gradients in the mixed layer are central to the upper ocean dynamics.

To derive the baroclinic shallow-water model (see Appendix A), we use a two-layer model as a starting point (e.g., Røed, 2012). We employ the rigid-lid approach ($\eta = 0$) for the surface elevation and decouple the upper layer from the lower layer by assuming that the latter is infinitely deep. Then, we obtain the so-called 1.5-layer model for the upper ocean:

$$\begin{bmatrix} \zeta \\ \widetilde{hu} \\ \widetilde{hv} \end{bmatrix}_t + \begin{bmatrix} \widetilde{hu} \\ \widetilde{hu^2} + \frac{1}{2}g'\widetilde{h}^2 \\ \widetilde{huv} \end{bmatrix}_x + \begin{bmatrix} \widetilde{hv} \\ \widetilde{huv} \\ \widetilde{hv^2} + \frac{1}{2}g'\widetilde{h}^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ f\widetilde{hv} \\ -f\widetilde{hu} \end{bmatrix} + \begin{bmatrix} 0 \\ \widetilde{\tau}_u \\ \widetilde{\tau}_v \end{bmatrix} + \begin{bmatrix} 0 \\ -\widetilde{r}\widetilde{u}\sqrt{\widetilde{u}^2 + \widetilde{v}^2} \\ -\widetilde{r}\widetilde{v}\sqrt{\widetilde{u}^2 + \widetilde{v}^2} \end{bmatrix}. \tag{2.3}$$

Here, and as shown in Figure 2.1, $\widetilde{h}$ is the MLD, given as $\widetilde{h} = \widetilde{H} + \zeta$, where ζ is positive downwards and the deviation from equilibrium MLD $\widetilde{H}$. Furthermore, $\widetilde{\tau} = [\widetilde{\tau}_u, \widetilde{\tau}_v]$ is the wind forcing, and $\widetilde{r}$ is the friction coefficient between the two layers. Most importantly, g' is the reduced gravity, given by

$$g' = \frac{\rho_2 - \rho_1}{\rho_2} g. \tag{2.4}$$

Note that (2.3) is also called reduced-gravity model and has the same mathematical structure as the shallow-water model in (2.1), with the only difference being the physical meaning of the deviation from equilibrium depth (ζ vs. η) and the gravitational constant (g' vs. g). This means that the GPU-accelerated simulation framework (Brodtkorb and Holm, 2021) can be used directly to simulate baroclinic dynamics, as well as barotropic. A consequence of the reduced gravity and the relatively small values of $\widetilde{h}$, is that the CFL-conditions in (2.2) allows us to take larger time steps. Whereas the barotropic model needs to take time steps in the order of 1 s, the baroclinic model can run with time steps in the order of 1 min to 2 min, which results in significantly faster baroclinic simulations than barotropic simulations of the same grid and time range. Therefore, we can add the baroclinic mode very inexpensively (with regards to computational resources).

A drawback of our approach is that we obviously lose important elements of the dynamics that stem from the coupling between the barotropic and baroclinic modes. Perhaps most important in the context of short-term simulations is that the baroclinic mode phase speeds in our reduced-gravity model are only functions of the reduced gravity and the mixed layer depth, while in reality the phase speeds depend on the thicknesses of both layers, with increasing error as the lower layer becomes shallower, such as is often the case in the coastal zone. This issue can at least partially be addressed by perturbing the MLD and the reduced gravity separately, and we will show some examples of this approach in Section 4.

2.3 Trajectory Model

To predict drift trajectories, we combine independent simulations of the barotropic and baroclinic models. A drifter is then primarily advected by the simulated currents $[\overline{u}, \overline{v}]$ and $[\widetilde{u}, \widetilde{v}]$, together with a contribution from the wind $[u_w, v_w]$. The direct contribution from the wind (the windage) depends largely on the shape of the drifting object. In this work, we are mainly interested in objects that are mostly submerged, so that their drift trajectories primarily depend on the currents and only to a limited extent on the wind. We model

windage through a leeway coefficient ψ that is an object dependent fraction of the wind speed. Hence, a drifter with position (x_t, y_t) will be advected according to

$$\begin{aligned} x_{t+\Delta t} &= x_t + \Delta t \left(\overline{u}(x_t, y_t, t) + \widetilde{u}(x_t, y_t, t) + \psi u_w(x_t, y_t, t) \right), \\ y_{t+\Delta t} &= y_t + \Delta t \left(\overline{v}(x_t, y_t, t) + \widetilde{v}(x_t, y_t, t) + \psi v_w(x_t, y_t, t) \right). \end{aligned} \tag{2.5}$$

We model the trajectories integrated in the GPU-framework to avoid copying data from the GPU, which would have been a major bottleneck for performance. Furthermore, we also avoid large amounts of output files that would be needed as input for a separate offline trajectory model. In practice, due to the large differences in time-step sizes between the two models, we need to take many steps with the barotropic model for each time we evolve the baroclinic model. We advect the drifters with the same time step and frequency as the barotropic model while using the most recent baroclinic time step, as shown in Algorithm 1. Note that when looking up values for the currents u and v, we interpolate in space, but not in time. This is because the information from the previous time step is lost after we update the model with the second-order Runge-Kutta step.

Algorithm 1 Simulation of drift trajectories using a combination of the barotropic and baroclinic models.

$\widetilde{t}, \overline{t} \leftarrow 0$
Initialize drifter: $(x, y) \leftarrow (x_0, y_0)$
while $\widetilde{t}, \overline{t} < t_{\text{end}}$ **do**
  Evolve baroclinic simulator $\widetilde{t} \leftarrow \widetilde{t} + \widetilde{\Delta t}$
  while $\overline{t} < \widetilde{t}$ **do**
    Evolve barotropic simulator $\overline{t} \leftarrow \overline{t} + \overline{\Delta t}$
    Advect drifter:
    $x \leftarrow x + \overline{\Delta t} \left(\overline{u}(x, y, \overline{t}) + \widetilde{u}(x, y, \widetilde{t}) + \psi u_w(x, y, \overline{t}) \right)$
    $y \leftarrow y + \overline{\Delta t} \left(\overline{v}(x, y, \overline{t}) + \widetilde{v}(x, y, \widetilde{t}) + \psi v_w(x, y, \overline{t}) \right)$
  end while
end while

2.4 Model Perturbations

We now look at ways to perturb initial conditions and external forcings from deterministic operational ocean forecasts. In our demonstration cases, the data is produced by ROMS-based 3D models, containing hourly time series of 3D currents, temperature, salinity, and sea-level, in addition to surface winds from the numerical weather model that forces the 3D ocean models.

Barotropic Initial and Boundary Conditions. For relatively small domains, such as those considered in this paper, the barotropic signals are mainly driven by the boundary conditions, and a common source of error in coastal areas is tidal phase. We therefore introduce perturbations in the barotropic models by timeshifting the initial and boundary conditions. Furthermore, drift trajectories can be sensitive to the location of eddies, and by timeshifting initial conditions we also get slight perturbations on the placements of eddies within the domain.

Mixed-Layer Depth and Reduced Gravity. The MLD can be defined in many different ways (e.g., Brainerd and Gregg, 1995; de Boyer Montégut et al., 2004; Giunta and Ward, 2022) and in our examples we choose a density criterion, where we compute the potential densities from temperature and salinity (Fofonoff and Millard, 1983) read from the 3D operational model. We then choose an initial MLD based on an isopycnal surface with a vertical position that we deem to be overall most representative of the pycnocline in our model domain. In some grid cells, the MLD may not be well-defined. We extrapolate from the closest "valid" grid cells in such cases. The variations in the value of the density criterion is used to perturb the baroclinic model

ensemble. As already mentioned, we expect errors in the phase speed of the baroclinic modes, hence we use the possibility to perturb the reduced gravity g' independently from the MLD perturbations.

Wind Field. In this work, we obtain hourly deterministic wind fields from the operational ocean models. We then perturb the wind influence onto the models by rotating the entire wind field by some angle. A more realistic way to obtain uncertain wind fields would be to use the ensemble of wind fields from a probabilistic atmospheric forecasts, if available.

Wind Stress. We calculate the wind stress $\bar{\tau}$ using the approach from Large and Pond (1981), which we apply directly to the depth-averaged currents in the barotropic model. However, upper ocean layers have a more direct response to wind forcing than depth-averaged currents. In particular with thin mixed layers, we expect that large momentum fluxes from the atmosphere are partially communicated instantaneously further down to the deep ocean through the shear stress between the layers. This could be obtained by locally increasing the inter-layer friction coefficient, but to increase the freedom of the reduced-gravity model, we instead fix a friction coefficient $\tilde{r}$ for the entire domain and reduce the wind stress $\tilde{\tau} = \tau_0 \bar{\tau}$ by an uncertain factor $\tau_0 \leqslant 1$.

Frictional Forces at the Base of the MLD. The friction in the reduced-gravity model represents the flux of momentum between the mixed layer and the ocean interior from shear stresses, near-inertial waves, and so on. As just mentioned , this is another way to reduce momentum in the upper layer, where frictional forces however depend on an acceleration of the water column. It may feels redundant but having both parametrizations gives more flexibility in the modelling. We do not consider any details of these mechanisms here and simply consider them parameterized within the uncertain friction coefficient $\tilde{r}$.

Numerical Drifters. In a realistic case of trajectory modeling in support of, e.g., search-and-rescue operations, the initial position is likely to be a major source of uncertainty. Since our aim here is to explore the uncertainties associated with the modeling systems, we do not perturb initial positions of our numerical drifters in our examples. We perturb the windage, through the leeway coefficient, to compensate for unresolved near-surface vertical shear in the currents, using distributions with high probability for $\psi \in [0.01, 0.05]$ (e.g., Sutherland et al., 2020), which is randomly assigned to each numerical drifter.

2.5 Combined Drifter Ensembles

Since the two simplified ocean models from Sections 2.1 and 2.2 are not coupled, we can run multiple barotropic and baroclinic simulations independently. A cross product of barotropic and baroclinic simulations spans out a large set of ocean currents that can be used for trajectory modeling. In principle, if we have an ensemble of N_{bt} barotropic models and an ensemble of N_{bc} baroclinic models, we can combine any pair of realizations from each of the ensembles to obtain $N_{bt} \times N_{bc}$ combined ocean states. Since we model passive drifters that do not influence the ocean state, we can also advect drifters with different leeway coefficients within each combined ensemble member. With N_ψ different leeway coefficients, we forecast the location of drifting objects based on $N_{bt} \times N_{bc} \times N_\psi$ simulated drift trajectories.

2.6 Initialization from 3D Operational Models

In all numerical experiments presented in this paper, we assume that we have a hourly results from a deterministic 3D ROMS model. For simplicity, we initialize and run our simplified models using the same horizontal grid resolution as the 3D model, meaning that landmask and bathymetry data is directly available.

The barotropic models is straightforward to initialize. Most often, η, $\tilde{H}$, $\bar{u}$ and $\bar{v}$ are available directly from the 3D simulation results and can be used directly. However, if $\bar{u}$ and $\bar{v}$ are not available, it is a simple procedure to obtain $\overline{hu}$ and $\overline{hv}$ by numerical integration of u and v, respectively, through the water column in every horizontal grid cell.

For the reduced-gravity models, we define $\widetilde{H} \equiv 0$ and initialize ζ according to the isopycnal obtained from a suitable MLD criterion. We then approximate the baroclinic currents by subtracting the barotropic current and integrating vertically, as

$$\widetilde{hu} = \int_{-\zeta}^{0} u - \bar{u} \, \mathrm{d}z \quad \text{and} \quad \widetilde{hv} = \int_{-\zeta}^{0} v - \bar{v} \, \mathrm{d}z, \tag{2.6}$$

for each grid cell in the domain. To compute the reduced gravitational constant from (2.4), we first compute the potential densities[2] for each point in the 3D grid. Then, we let ρ_1 and ρ_2 be the mean volumetric averages of the potential energies above and under the MLD, respectively.

3 Assessing the Baroclinic Model

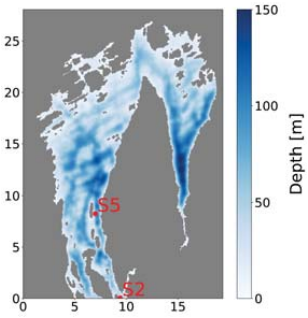
The barotropic part of the model from Section 2.1 is described in Holm et al. (2020a) and (Brodtkorb and Holm, 2021). Here we focus on evaluating the baroclinic model from Section 2.2. Coastal waves may have a significant influence on the dynamics in fjords through influencing the exchanges with the open ocean, and in some large fjord systems the spatial scales may allow for rotational effects on the internal wave dynamics. An idealized Kelvin wave test case is therefore provided in Appendix B. Here, we apply the reduced-gravity model to a high-resolution scenario in the inner Oslofjord, and compare our simulation results against a survey of tidally induced baroclinic currents (Staalstrøm et al., 2012), and assess the model's applicability through drift trajectories, short-term prediction, and computational performance.

3.1 Tidally Induced Baroclinic Currents in the Inner Oslofjord

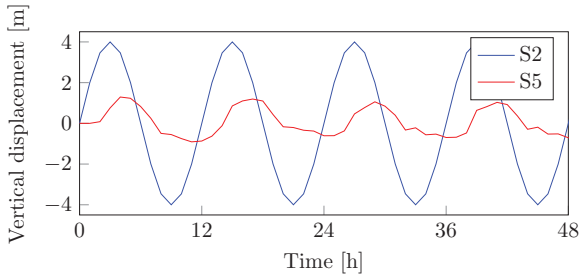
The inner and outer parts of the Oslofjord are separated by a shallow and narrow sill outside of the small city of Drøbak (see Figure C.1b). There are no significant freshwater sources in the inner Oslofjord, and the currents in the inner fjord are mainly driven by tidally induced baroclinic waves generated at the sill in the southern end, and the wind forcing. In the following, we will focus on the tidal forcing. We consider a $50\,\mathrm{m} \times 50\,\mathrm{m}$ uniform grid covering the inner Oslofjord, made from re-projecting data from the operational FjordOs model (Hjelmervik and Kristensen, 2019). The resulting 387×561 grid encompasses a bit less than a $20\,\mathrm{km} \times 30\,\mathrm{km}$ domain, as it can be seen for the bathymetry in Figure 3.1a.

Staalstrøm et al. (2012) presented a survey of the tide-induced upper layer dynamics in the inner Oslofjord during spring time. We use these results as reference to assess our reduced-gravity model. They refer to observation locations called "S2" just after the sill, where our computational domain starts, and "S5" about 7 km further into the fjord. Both locations are marked in Figure 3.1a. We model the boundary conditions to correspond to the observations made at location S2, and use a constant mixed-layer depth of 15 m for initial conditions (cf. Staalstrøm et al., 2012, Table 2). For the boundary conditions, we use the flow-relaxation scheme (Davies, 1976) at the sill, where an island separates the sill into a right and left branch. As external solutions, we employ a sine-shaped $\widetilde{v}$-current with amplitude $0.3\,\frac{\mathrm{m}}{\mathrm{s}}$ and a ζ-displacement with amplitude 4 m in the right branch. The values for the amplitudes are found in Staalstrøm et al. (2012, Figure 6b, 8a) and both waves have a period of 12 h. In the left branch, $\widetilde{v}$ is not forced from the boundary. The boundary conditions for $\widetilde{u}$ are kept zero everywhere. Moreover, Staalstrøm et al. (2012, Table 2) suggests a density difference of around $10\,\frac{\mathrm{kg}}{\mathrm{m}^3}$, leading to a reduced gravity $g' = 0.1\,\frac{\mathrm{m}}{\mathrm{s}^2}$. The friction coefficient is finally set to 0.0015 m and the simulation is run without wind forcing for a time span of two days. Our model results then produce signals with phase speeds of about $1.2\,\frac{\mathrm{m}}{\mathrm{s}}$, which is in good agreement with Staalstrøm et al. (2012, Section 3.1), and corresponding to a Rossby radius of a bit more than 10 km. Figure 3.1b shows the results of the reduced-gravity model, along with the boundary conditions in S2. The values for ζ at S5 have a significant reduced amplitude with maximum values at about 1 m, corresponding to about 25% of the amplitude at S5. The signal also lags in time. This is again in accordance with the observations in Staalstrøm et al. (2012, Section 3.2).

[2] This can be done in multiple ways. We have used the EOS80 module in the Python package `seawater` (Fernandes, 2015), which implements Fofonoff and Millard (1983) for the purpose.



(a) Bathymetry of inner Oslofjord with locations "S2" and "S5", and axes in km.



(b) Vertical displacement of ζ at locations "S2" and "S5". Note that S2 essentially describes the set boundary conditions.

Figure 3.1: The inner Oslofjord with selected observation locations from Staalstrøm et al. (2012) marked by red dots and the simulated vertical displacement in these locations.

3.2 Baroclinic Contributions to Drift Trajectories

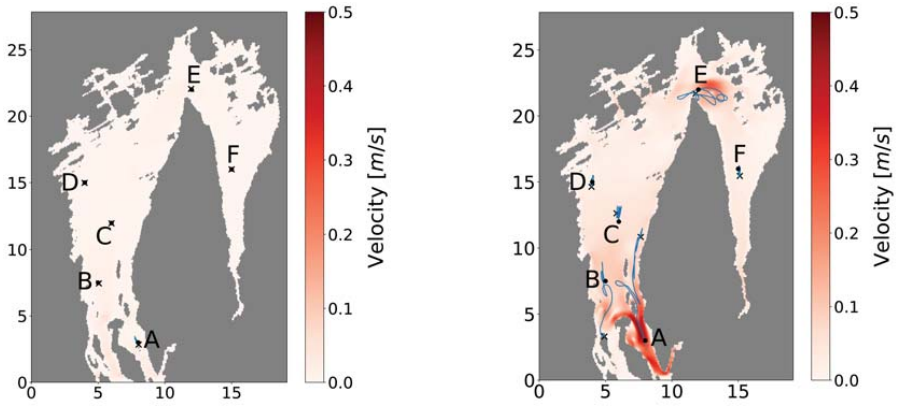
To evaluate the roles of the baroclinic and barotropic modes, we also run the barotropic model with idealized conditions. Again, we initialize all variables to zero, and force with tidal signals of up to 0.1 $\frac{\mathrm{m}}{\mathrm{s}}$ current speed and 0.15 m displacement, in accordance to typical values from FjordOs. We demonstrate the application of Algorithm 1 for a 48 h simulation.

Figure 3.2 shows the resulting drift trajectories for a set of six drifters, where the initial drifter positions are annotated with letters, (a) shows trajectories using only the barotropic currents, and (b) shows trajectories using the combined currents as described in Section 2.3. The crosses mark the final drifter positions. We note that the drifters seeded in the barotropic model barely move at all. In contrast, drifters using combined current information travel longer distances. For them, the main contribution to the drift comes from the baroclinic model. Some drifters are mostly advected back and forth by the tidally induced currents, while other trajectories are more irregular or persistent in certain directions.

Our baroclinic model and method of initialization are simple, and to illustrate some of the challenges associated with capturing the non-barotropic flows, we initialize two baroclinic model fields from the operational hindcast produced by the FjordOs model using data separated by one hour. We then run the simulation initialized by the earliest point in time for one hour to catch up with the other, and compare these currents. Figure 3.3 shows the results. Here, we have used MLD criterion $\rho = 1023 \frac{\mathrm{kg}}{\mathrm{m}^3}$ and no friction. The main dynamical features are reproduced in our reduced-gravity model, e.g. the jet entering the fjord from the sill at the southern boundary, the eddy in the widest section of the fjord, and the current bending around the headland in the northern part of the domain. The initial state contains imbalances, however, leading to some of the energy being lost to noise, and it is evident that the overall strength of the flow is reduced.

3.3 Computational Performance

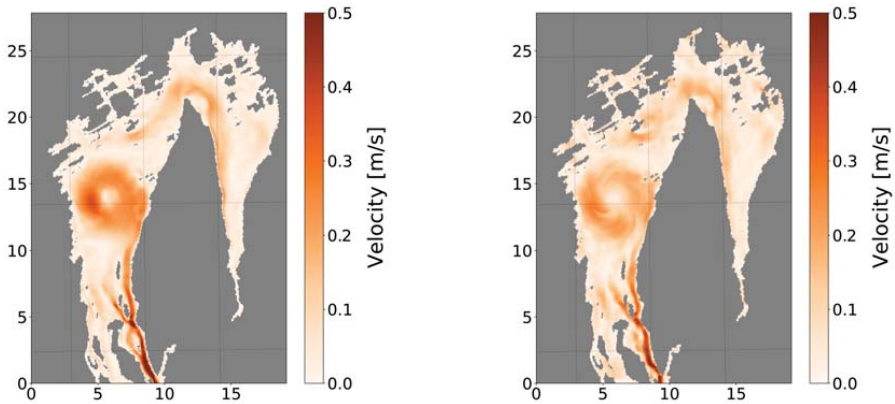
A comparison of the wall run times provides us with estimates of how many ensemble members we can afford to run for the same computational costs as a full 3D simulation. For the idealized barotropic and baroclinic simulations, the computational cost is assessed as the wall time which the numerical scheme requires to simulate one hour on a NVIDIA GeForce RTX 3090 using 0.8 as Courant number. Instead of comparing directly to FjordOS, which covers in its full extent a larger domain with a curvi-linear grid, we consider two independent idealized ROMS setups with the same spatial extent as the GPU models. One of the



(a) Barotropic drift trajectories

(b) Trajectories for combined currents

Figure 3.2: Drift trajectories for different simplified models under idealized conditions. The background shows particle velocity after 48 h, and the axes are in km.



(a) FjordOS

(b) Baroclinic model

Figure 3.3: Absolute velocities of the currents in the barotropic model in the Oslofjord at the same point in time and with the same MLD criterion. The left panel shows the currents as initialized directly from FjordOs, whereas the right panel was initialized one hour earlier and simulated for one hour.

Table 3.1: Wall run times to simulate 1 h in the inner Oslofjord with different models.

	ROMS (2 nodes/4 CPUs/64 cores)		GPU Ocean (Single GPU)	
	Full 3D	**Barotropic 2D**	**Barotropic 2D**	**Baroclinic 2D**
Run time	923.1 s	53.1 s	2.65 s	0.083 s
Relative to GPU Ocean barotropic	349.0	20.1	1	0.031
Relative to GPU Ocean baroclinic	11,122.3	639.4	31.9	1

ROMS setups was run in full 3D mode (including both barotropic and baroclinic physics) with ten vertical layers, denoted as *Full 3D*, and one in 2D mode (only utilizing the barotropic physics of ROMS), denoted as *Barotropic 2D*. These simulations were run on two nodes with a total of 64 cores divided on four Intel Xeon Gold model 6130 (2.1 GHz) CPUs, connected with Intel OmniPath (100 Gbit/s).

Table 3.1 summarize the results, where we see that the ROMS models on 64 CPU cores with a total CPU time of more than 16 min for full 3D simulation, and just below 1 min for the barotropic 2D simulation. We note that overhead due to MPI communication in the ROMS model simulations is about 15 and 30 percent, respectively, of the total run time for the full 3D and barotropic 2D test cases. This is about 350 times longer for the full 3D ROMS simulation, and about 20 times longer for the barotropic 2D ROMS simulation than the GPU-accelerated barotropic model, which again requires about 32 times longer to run compared to the GPU-accelerated baroclinic model. For our two simplified models, the CFL restrictions depend on the phase speeds $c_{\max} \approx \sqrt{g h_{\max}}$, which for these cases have a ratio at about 32.6.

4 Ensemble Prediction for Drift Trajectories

We will now use the simplified models to make short-term ensemble predictions for drift trajectories in three different areas in Norway, see Figure C.1a for details. These cases demonstrate how the barotropic and baroclinic processes contribute to a varying degree to the drift in different areas, and showcase how a standard barotropic shallow-water model, a reduced-gravity model, or a combination of both can serve as simplified models for drift predictions. In all three cases, we use simulation results from ROMS 3D models for initial and boundary conditions, and carry out sensitivity studies to identify how the different parameters in the models contribute to ensemble spread. We determine parameter ranges for the ensemble simulations by looking at the variances in the velocity field throughout each domain and the spread in selected drift trajectories with respect to the individual parameters. Moreover, we use OpenDrift (Dagestad et al., 2018) to produce reference trajectories using the uppermost layer from the ROMS 3D model simulations. Within OpenDrift, we simply use the OceanDrift model, and we generate references using three different leeway coefficients.

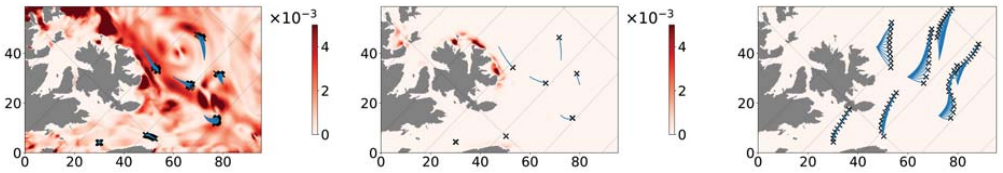
The operational ocean forecasts from MET Norway are made with the Norkyst-800 modeling system (Albretsen et al., 2011), which is based on the ROMS (Shchepetkin and McWilliams, 2005) general circulation model. Herein, we use the NorFjords-160 and FjordOs model systems as our references, both of which are one-way nested into Norkyst-800. NorFjords-160 is a collection of 13 model domains that cover the entire Norwegian coastline with a grid resolution of 160 m each. FjordOs is an operational high-resolution ROMS-model for the Oslofjord as described in Røed et al. (2016) where the authors also highlight the benefits of high spatial resolution for complex topographies. It uses a curvilinear grid with horizontal resolution within the inner Oslofjord of 50–100 m. NorKyst-800 forecasts[3] and FjordOs hindcast[4] are available through MET Norway's THREDDS Data Server, and NorFjords data is freely available upon request.

[3] `https://thredds.met.no/thredds/fou-hi/norkyst800v2.html`

[4] `https://thredds.met.no/thredds/catalog/fjordos/fjordos2_hindcast/v6/exp1/catalog.html`

Table 4.1: Uncertain parameters and their perturbations as used in the barotropic sensitivity study and the trajectory forecasts outside of North Cape.

Sim	Parameter	Fixed value	Uncertainty spans (sensitivity study)		(trajectory forecast)	
Barotropic	initial conditions	10:00	[00:00, 17:00]	18	[00:00, 17:00]	6
Barotropic	wind rotation	0°	$[-45°, 45°]$	19	$[-10°, 10°]$	5
Drift advection	windage, ψ	0.0	$[0, 0.05]$	11	$\mathcal{N}_+(0.03, 0.015)$	10



(a) Timeshifted initial conditions (b) Wind rotation (c) Windage

Figure 4.1: Sensitivity to varying individual parameters in the barotropic model. The background fields in (a) and (b) show variance in water velocity, and all axes are in km.

In the following, we initialize ensembles of simplified models using the same grid resolution as NorFjords-160 for North Cape and the Boknafjord, and project the FjordOs curvilinear gridded data onto a 50 m uniform grid for the Oslofjord. To illustrate that the ensembles are light-weight, all experiments are carried out on a single GPU and we use the available GPU memory as an upper constraint on the ensemble size. To further increase ensemble size or computational speed, spreading the ensemble across multiple GPUs using MPI is a viable option (Holm et al., 2020b).

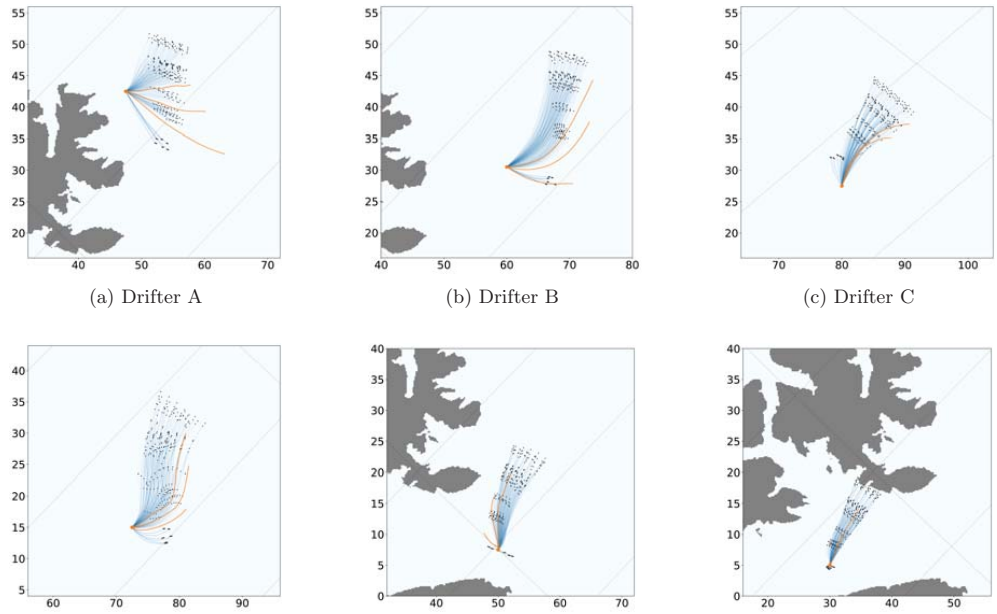
4.1 North Cape

The first example showcases the shallow-water model as described in Section 2.1. In the open waters of the Barents Sea, density anomalies are typically small and barotropic dynamics often dominate. Hence, we are unable to initialize reduced-gravity models based on our framework in this area, and therefore construct an ensemble of barotropic simplified models only. We focus on a chosen area just outside of North Cape. The reference initial conditions together with the initial placement of drifters can be seen in Figure C.2a.

4.1.1 Barotropic Parameter Sensitivity Study

We start by analyzing the sensitivity of the model result with respect to perturbations of each parameter based on our discussion in Section 2.4. To do so, we run a set of simulations with equidistant values for the individual uncertain parameters within their respective ranges, while keeping the other parameters fixed. Table 4.1 lists the uncertain parameters and their values. We then evaluate the sensitivity to the model results by looking at the variance in water velocity and the spread in the obtained drift trajectories after six hours. Note that we use no windage as a reference to more clearly investigate how the other uncertain parameters affect the ocean state directly.

Figure 4.1 shows the results of the barotropic sensitivity analysis. Note that there is an eddy centered approximately at $(60, 40)$ in the figures. With time-shifted initial conditions in Figure 4.1a, the spread in water velocity is especially strong around the eddy and at the upper boundary where tidal signals enter the domain. In contrast, Figure 4.1b shows that the wind response is too slow to generate significant spread in the ocean states. If we use windage with leeway coefficient $\psi > 0$, however, the different wind directions will give a larger spread in the trajectories through the direct wind drag on the drifters such that we still



(a) Drifter A (b) Drifter B (c) Drifter C

(d) Drifter D (e) Drifter E (f) Drifter F

Figure 4.2: Barotropic drifter ensemble simulations for selected drifters. The orange trajectories are deterministically calculated using the the surface currents from NorFjords and OpenDrift with leeway factors 0%, 1.5% and 3%. All axes are in km.

want to include a reasonable number of wind directions into a drift ensemble. This behavior is illustrated by Figure 4.1c, where we see the large effect of windage for a non-rotated wind field.

4.1.2 Barotropic Drift Trajectory Simulations

For the actual drifter experiment, we build an ensemble consisting of 30 barotropic models by mixing six different initial conditions and five wind rotations. Note that we have reduced the uncertainty in the wind direction here, compared to the values used in the sensitivity study. In addition, we add drifters with ten different leeway coefficients to each ensemble member, which are generated as positive realizations from a Gaussian distribution. Thus, we produce 300 drift trajectories in total.

Figure 4.2 shows the resulting drift trajectories. For drifter A and B, the ROMS reference lies outside the ensemble, while for all other starting positions they are covered by the ensemble spread and with the exception of drifter A, all paths in the ensemble follow similar shapes as the references. In general, the ROMS trajectories are longer than those from our ensemble, which is due to the surface currents being stronger than our depth averaged currents. It is clear that our barotropic model does not capture all dynamical features relevant for the drift. The currents in the ROMS 3D model are not entirely barotropic, as can be seen in Figure C.2. In spite of these shortcomings, the barotropic ensemble modeling results are quite good. We recognize a clustering in the trajectories of drifter A, which is due to the leeway factors. This effect is less pronounced in the other cases, where the wind direction is not orthogonal to the current direction. Drifter F is mostly wind driven, since the currents at the entrance of the Porsangerfjord are weak, and we note that the ROMS reference lies in the middle of the ensemble.

Table 4.2: Uncertain parameters and their perturbations used in the baroclinic sensitivity study and trajectory forecast for the inner Oslofjord.

Sim	Parameter	Fixed value	Uncertainty spans (sensitivity study)		(trajectory forecast)	
Baroclinic	wind rotation	$0°$	$[-45°, 45°]$	19	$[-10°, 10°]$	5
Baroclinic	wind stress factor, τ_0	0.75	$[0.1, 1.0]$	10	$\mathcal{N}_+(0.35, 0.2)$	36
Baroclinic	inter-layer friction, $\tilde{r}$	0.003	$[0, 0.005]$	11	$\mathcal{N}_+(0.0025, 0.001)$	
Baroclinic	MDL criterion, ρ	1024.0	$[1022.0, 1024.5]$	6	$\mathcal{U}[1022.5, 1024.5]$	
Baroclinic	reduced gravity from MDL, g'	0.023	$[0.020, 0.032]$	6		
Baroclinic	reduced gravity set directly, g'	0.023	$[0.01, 0.10]$	10		
Drift advection	windage, ψ				$\mathcal{N}_+(0.03, 0.015)$	10

4.2 Inner Oslofjord

As we saw in Section 3, baroclinic dynamics dominates in the inner Oslofjord, with minimal contribution from the barotropic currents. As a consequence, we use an ensemble of reduced-gravity models only to predict drift trajectories in this area, and we start again by investigating how much influence each uncertain parameter has on the resulting ocean states of the reduced-gravity model and the drift trajectories.

4.2.1 Baroclinic Parameter Sensitivity Study

Table 4.2 shows the parameters of the baroclinic model that we perturb. As discussed in Section 2.4, the initial MLD and the value of the reduced gravitational constant g' are coupled, but we evaluate the model's sensitivity to perturb g' alone while keeping the initial MLD fixed, and to perturb the MLD alone with g' fixed, since we do introduce errors in the signal phase speeds by decoupling the modes. Moreover, we test the model sensitivity to the inter-layer friction, wind directions, and wind stress factors. Figure 4.3 shows the potential density of the upper 25 m along a south-north cut, along with five different isopycnals. The pycnocline is generally above 10 m depth.

Figure 4.4 show the result of the sensitivity analysis for the reduced-gravity model. The wind direction has limited influence on the circulation. The winds in this case are moderate, and the wind stress perturbation leads to a modest spread of the drifters in the wind direction. The perturbation of the interface friction causes the largest variations in the ensemble. Perturbations of the MLD and the reduced gravity have limited influence on ensemble spread, with little difference between the coupled and decoupled treatment of the two parameters.

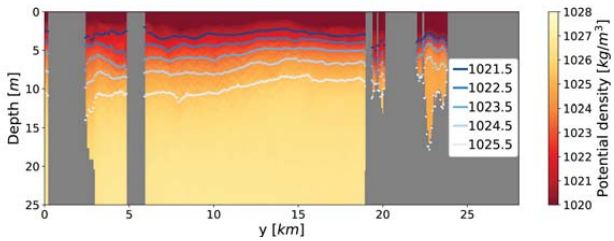
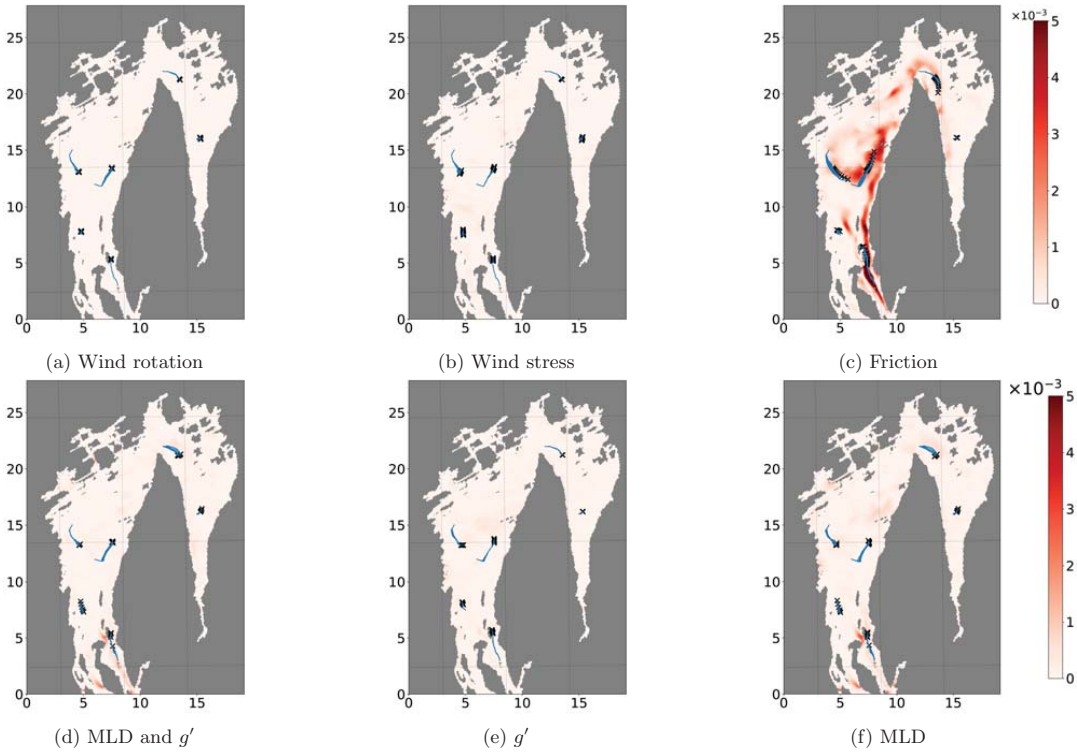


Figure 4.3: Isopycnals for a cross section along the y-axis through the strait of the 'S5'-location in the Oslofjord (see Figure 3.1a for location).



(a) Wind rotation (b) Wind stress (c) Friction

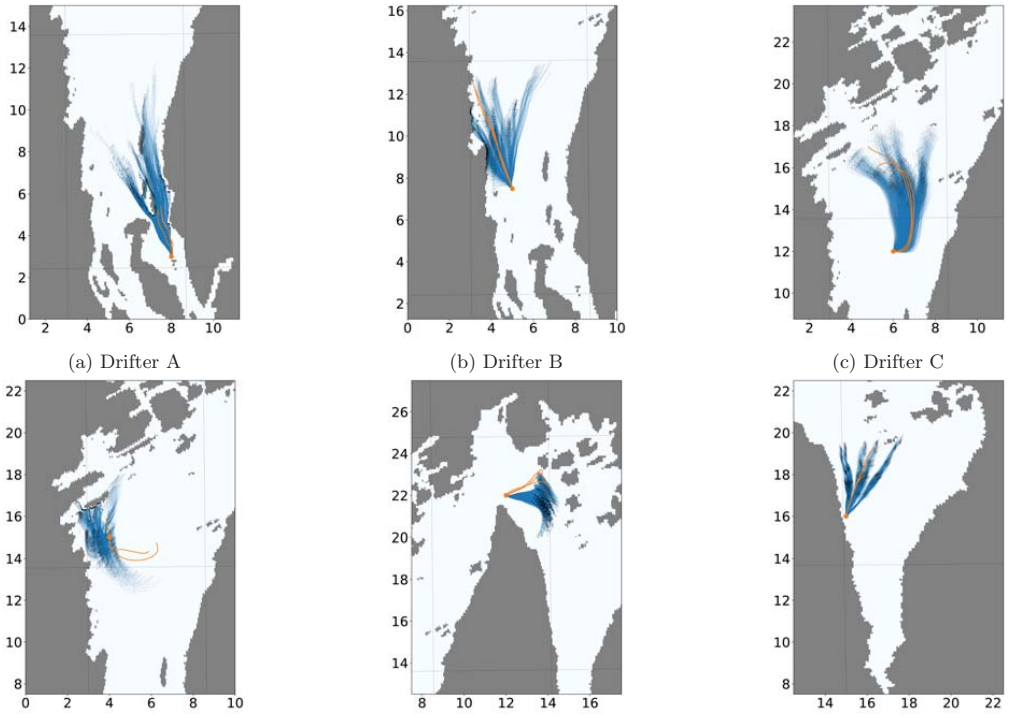
(d) MLD and g' (e) g' (f) MLD

Figure 4.4: Sensitivity to varying individual parameters in the reduced-gravity model. The background fields show variance in water velocity, with axes in km.

4.2.2 Baroclinic Drift Trajectory Simulations

From the baroclinic parameter sensitivity study, we infer reasonable parameter distributions for the construction of a baroclinic ensemble, as summarized in Table 4.2. We neglect the decoupled sampling of MLD and g', since it results in little variability, and will only use the coupled MLD and g'. For wind stress, interlayer friction and MLD criterion, we now sample values from positive Gaussian and uniform distributions, in contrast to the parameter study where we fixed equidistant values within selected ranges. For the wind rotation, however, we still use fixed equidistant values, due to technical constraints in how the wind forcing is implemented in the GPU framework. We take advantage of the fact that the reduced-gravity models runs significantly faster than the barotropic models, and use a relatively large ensemble. In total, we consider 270 baroclinic simulations and ten leeway factors per instance.

In Figure 4.5 we show the ensemble trajectory predictions for each drifter. For drifter A, all three reference trajectories from ROMS strand eventually, together with a big share of the ensemble members. For drifters B, C and F, the ensemble spread is dominated by the spread in the wind direction, but lies around the reference trajectories. The drifter D trajectories bifurcates, and interestingly this is evident in both the ensemble and the ROMS reference simulations. For drifter E, the reference simulations fall outside of the ensemble, which is due to unresolved near-surface vertical velocity shear in the reduced-gravity model ensemble.



(a) Drifter A (b) Drifter B (c) Drifter C

(d) Drifter D (e) Drifter E (f) Drifter F

Figure 4.5: Baroclinic drifter ensemble simulation zoomed in on each individual drifter. Orange trajectories come from OpenDrift using FjordOs surface currents and leeway factors 0%, 1.5% and 3%. The axes are in km.

4.3 Boknafjord

The third example is from the Boknafjord just north of Stavanger. Here we find that both barotropic and baroclinic currents play a significant role for the drift, and the drift predictions are based on adding currents from the barotropic and baroclinic models. This is shown by Figure 4.6, where we have plotted deterministic trajectories using the barotropic, the baroclinic, and the combined model for four drifters. In particular, drifter D has a 180° difference between the trajectories from the two individual models, and for drifters B and D, this difference is on 90°. This underlines that neglecting either of the signals results in a loss of information. By using the combined current from both simplified models, we get deterministic trajectories that lie in between the individual models.

4.3.1 Parameter Sensitivity Study

Similarly to the two former cases, we carry out a parameter sensitivity study. Table 4.3 lists the relevant parameters, where the most important differences from the previous cases are the sensitivities related to the initial MLD. The pycnocline in the Boknafjord is well defined, but the stratification is weaker than in the Oslofjord. Hence, small changes in the MLD density criterion significantly change the resulting depth of the mixed layer. The span of the densities for the MLD is therefore here adapted to a smaller interval.

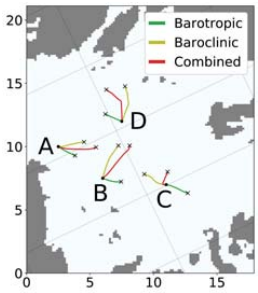


Figure 4.6: Deterministic drift trajectories calculated with different simplified models and leeway coefficient $\psi = 0$. Axes are in km.

Table 4.3: Uncertain parameters and their perturbations used in the combined sensitivity study and trajectory forecast in the Boknafjord.

Sim	parameter	uncertainty sampling	
Barotropic	wind rotation	$[-10°, 10°]$	5
Barotropic	initial condition	$[05:00, 17:00]$	3
Baroclinic	wind rotation	$[-10°, 10°]$	5
Baroclinic	wind stress factor, τ_0	$\mathcal{N}_+(0.35, 0.25)$	36
Baroclinic	inter-layer friction, $\widetilde{r}$	$\mathcal{N}_+(0.0025, 0.001)$	
Baroclinic	MDL criterion, ρ	$\mathcal{U}[1022.75, 1023.5]$	
Drift advection	windage, ψ	$\mathcal{N}_+(0.03, 0.015)$	10

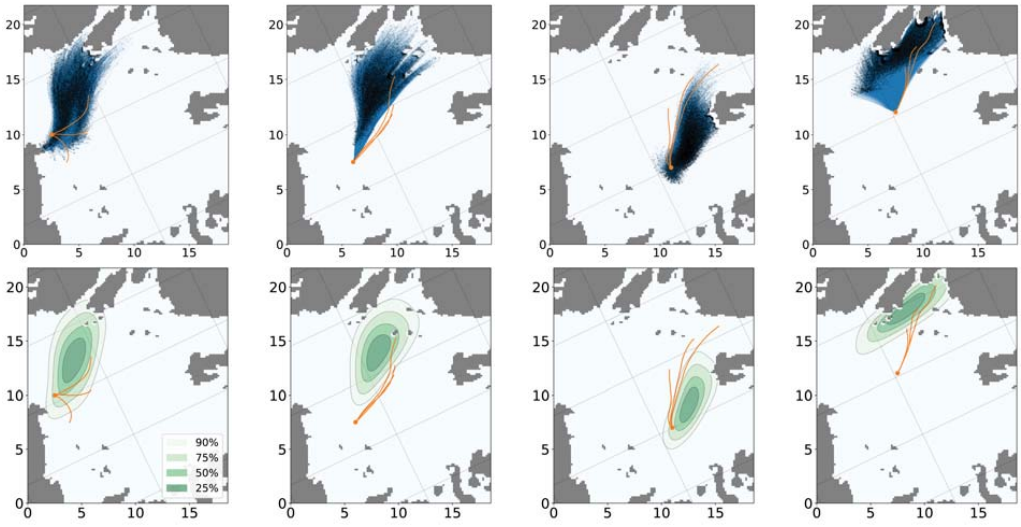
Furthermore, a shallow MLD responds more strongly to variations in the wind, which means that the effects of the MLD perturbations on the drift are not expected to be symmetrically distributed around the MLD average. The MLD is in general more shallow in the south-western part of the domain (left side of the plots) and deeper in the north (top right in the plots).

4.3.2 Combined Drift Trajectory Simulations

We use ensembles of 15 barotropic and 180 baroclinic members, respectively. In addition, we sample ten leeway coefficients per member, which results in 27,000 combined trajectories in total for each of the four drifters. Figure 4.7 shows the resulting forecasted trajectories, where the top row shows all individual trajectories and the bottom row shows the probability distribution through estimated kernel densities for the drifters' final positions after six hours. It should be noted that the wind direction without rotation is roughly parallel to the y-axis. Since the currents of the barotropic and baroclinic model in general are not aligned, we get a noticeable increase in the trajectory spread. Furthermore, we see that for drifters B and C in particular, the reference trajectories based on the ROMS model and OpenDrift go in the direction of the deterministic combined trajectories from Figure 4.6. By looking at the ensemble trajectories, we see that the characteristics of the reference trajectories are represented in the ensembles. . For all drifters except for drifter C, the reference trajectories are within the 90% confidence area of the ensemble distribution, illustrating the potential of the combination of our approach.

5 Discussion

The starting point of this paper was a GPU-accelerated framework for running ensembles of barotropic simplified ocean models intended for short-term drift trajectory predictions in real-world domains based on



(a) Drifter A (b) Drifter B (c) Drifter C (d) Drifter D

Figure 4.7: Ensemble drift forecast made from the combined simplified model in the Boknafjord. The trajectories are shown at the top, and the probability distribution after six hours at the bottom. Orange trajectories are computed by OpenDrift using NorFjords surface currents and leeway factors 0%, 1.5% and 3%. The axes are in km.

the shallow-water equations. In this paper, we have presented how the framework can be expanded to also run ensembles of baroclinic reduced-gravity models by reusing the underlying computational code. Furthermore, since the phase speeds of reduced-gravity models are much smaller than their barotropic counterparts, we can increase the temporal step sizes of the models significantly which give increased computational performance. We have also proposed to simulate drift trajectories using a combination of the barotropic and reduced-gravity models, and shown how we can form large drift trajectory ensembles by combining the simulated currents from any pair of models from an ensemble of barotropic models and an ensemble of reduced-gravity models. To demonstrate the use of both barotropic and baroclinic simplified models, we presented drift trajectory simulations for three different areas along the Norwegian coast. The examples show barotropic-dominated currents (North Cape), currents dominated by baroclinic dynamics (Oslofjord), and contributions from both (Boknafjord). We also discussed the sensitivity to uncertain parameters for all cases and presented several ensemble simulations for a range of numerical drifters with different initial positions. Our results show that the ensemble spread generated by the simplified models most often cover the deterministic trajectories we used as reference solutions, which are computed from the surface currents from a 3D general circulation model.

In a real-world setting, the underlying assumptions on which the simplified models are based are certainly not always fulfilled. The self-imposed constraint of reusing very efficient existing numerical solvers forces us to make compromises with regard to model complexity and accuracy. There are issues with defining the MLD, especially in cases without a well-defined pycnocline, and in shallow waters where the assumption of a deep lower layer is clearly invalid, we cannot expect the reduced-gravity model to perform well. We want to reiterate, however, that our primary aim is not to replace full 3D circulation models and provide accurate descriptions of the state of the ocean. Our goal is rather to use the simplified models to provide additional tools for decision support when short term forecasts of surface drift are needed. Our main region

of interest is the Norwegian coastal zone, which is complex both in terms of topography and freshwater forcings. In most cases of practical interest, incidents happen close to land where the main source of error in our conventional ocean forecasting system is lack of spatial resolution. With the numerical tools presented here, it is straightforward to spawn large ensembles of simplified models with very high resolution, combining robust physics for the barotropic dynamics with a more rudimentary description of the MLD dynamics, but which at least provide the opportunity to estimate the lowest order baroclinic mode.

For drift trajectory modeling, the input data from ocean models contributes significantly to the total uncertainty, and it is our hope that the simplified models we present here can be integrated as additional tools in the operational production chain. It remains, of course, to study carefully how these tools perform under a variety of environmental conditions so that their use is limited to cases where they add value. This also includes a more sophisticated and robust analysis of the uncertainty ranges for creating realistic ensemble spread. In this study, we have used very simple models for the complex coastal ocean dynamics that are better captured by fully 3D circulation models. As mentioned in the introduction, our simplified models need only be reasonably accurate on the short timescales relevant to decision support and in situations that are suitable for rapid update cycling, but more research is needed on better initialization procedures and improved model physics, ideally using coupled modes if efficient implementations can be made on heterogeneous computing platforms. Finally, we would like to stress that large ensembles of these low-dimensional simplified models are particularly well suited for fully nonlinear data assimilation methods (Van Leeuwen, 2009; Holm et al., 2020b), which is relevant for accurate short term drift forecasting in situations where the position of the object is known from, e.g., GPS-trackers.

Acknowledgement. This work is supported by the Research Council of Norway under grant number 310515 (Havvarsel).

Code Availability. The source code used to produce the results presented in this paper is available under a GNU free and open source license in order to enhance scientific exchange. The numerical results can be reproduced using Jupyter notebooks at `https://github.com/FlorianBeiser/reduced-gravity-ocean-model`. The simplified models and GPU-accelerated framework can be found on `https://github.com/metno/gpuocean`, and the results are made using the version with commit hash `https://github.com/metno/gpuocean/tree/711902d818931f354021a809b5b19626dc97837e`.

Disclosure Statement. The authors report that there are no competing interests to declare.

References

Albretsen, J., Sperrevik, A., Staalstrøm, A., Sandvik, A., Vikebø, F., and Asplin, L. (2011). NorKyst-800 Report No. 1 User Manual and technical descriptions. *Fisken og Havet*, 2:1–48.

Arango, H. and Reid, R. (1991). A generalized reduced-gravity ocean model. *Atmosphere-Ocean*, 29(2):256–287.

Beiser, F., Holm, H., and Eidsvik, J. (2023). Comparison of ensemble-based data assimilation methods for sparse oceanographic data. *preprint: arXiv:2302.07197, [accepted for publication in Quarterly Journal of the Royal Meteorological Society]*.

Bleck, R. (2002). An oceanic general circulation model framed in hybrid isopycnic-Cartesian coordinates. *Ocean Modelling*, 4(1):55 – 88.

Brainerd, K. and Gregg, M. (1995). Surface mixed and mixing layer depths. *Deep Sea Research Part I: Oceanographic Research Papers*, 42(9):1521–1543.

Brodtkorb, A. and Holm, H. (2021). Coastal ocean forecasting on the GPU using a two-dimensional finite-volume scheme. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 73(1):1–22.

Brodtkorb, A., Sætra, M., and Altinakar, M. (2012). Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. *Computers and Fluids*, 55:1–12.

Castro-Díaz, M., Fernández-Nieto, E., González-Vida, J., and Parés-Madroñal, C. (2011). Numerical treatment of the loss of hyperbolicity of the two-layer shallow-water system. *Journal of Scientific Computing*, 48(1):16—-40.

Chassignet, E., Smith, L., Halliwell, G., and Bleck, R. (2003). North Atlantic simulations with the Hybrid Coordinate Ocean Model (HYCOM): Impact of the vertical coordinate choice, reference pressure, and thermobaricity. *Journal of Physical Oceanography*, 33(12):2504–2526.

Chertock, A., Dudzinski, M., Kurganov, A., and Lukáčová-Medvid'ová, M. (2018). Well-balanced schemes for the shallow water equations with Coriolis forces. *Numerische Mathematik*, 138(4):939–973.

Christensen, K., Breivik, O., Dagestad, K., Röhrs, J., and Ward, B. (2018). Short-term predictions of oceanic drift. *Oceanography*, 31(3):59–67.

Dagestad, K.-F., Röhrs, J., Breivik, Ø., and Ådlandsvik, B. (2018). OpenDrift v1.0: a generic framework for trajectory modeling. *Geoscientific Model Development*, 11(4):1405–1420.

Davies, H. (1976). A lateral boundary formulation for multi-level prediction models. *Quarterly Journal of the Royal Meteorological Society*, 102(432):405–418.

de Boyer Montégut, C., Madec, G., Fischer, A., Lazar, A., and Iudicone, D. (2004). Mixed layer depth over the global ocean: An examination of profile data and a profile-based climatology. *Journal of Geophysical Research: Oceans*, 109(C12).

de la Asunción, M., Mantas, J., and Castro, M. (2011). Simulation of one-layer shallow water systems on multicore and CUDA architectures. *The Journal of Supercomputing*, 58(2):206–214.

de la Asunción, M., Castro, M., Mantas, J., and Ortega, S. (2016). Numerical simulation of tsunamis generated by landslides on multiple GPUs. *Advances in Engineering Software*, 99:59–72.

Dellar, P. and Salmon, R. (2005). Shallow water equations with a complete coriolis force and topography. *Physics of fluids*, 17(10):106601.

Dong, T., Dobrev, V., Kolev, T., Rieben, R., Tomov, S., and Dongarra, J. (2014). A step towards energy efficient computing: Redesigning a hydrodynamic application on CPU-GPU. In *2014 IEEE 28th International Parallel and Distributed Processing Symposium*, pages 972–981. IEEE.

Dyakonova, T. and Khoperskov, A. (2018). Bottom friction models for shallow water equations: Manning's roughness coefficient and small-scale bottom heterogeneity. *Journal of Physics: Conference Series*, 973(1):012032.

Fernandes (2015). python-seawater v3.3.2.

Fofonoff, P. and Millard, R. (1983). Algorithms for computation of fundamental properties of seawater. *UNESCO technical papers in marine science*, (44).

Gent, P. and Cane, M. (1989). A reduced gravity, primitive equation model of the upper equatorial ocean. *Journal of Computational Physics*, 81(2):444–480.

Giunta, V. and Ward, B. (2022). Ocean mixed layer depth from dissipation. *Journal of Geophysical Research: Oceans*, 127(4).

Gottlieb, S. and Shu, C.-W. (1998). Total variation diminishing Runge-Kutta schemes. *Mathematics of Computation*, 67(221):73–85.

Hjelmervik, K. and Kristensen, N. (2019). The Oslo fjord ocean model: communication and applications. In *OCEANS 2019-Marseille*, pages 1–5. IEEE.

Holm, H., Brodtkorb, A., Broström, G., Christensen, K., and Sætra, M. (2020a). Evaluation of selected finite-difference and finite-volume approaches to rotational shallow-water flow. *Communications in Computational Physics*, 27(4):1234–1274.

Holm, H., Sætra, M., and Brodtkorb, A. (2020b). Data assimilation for ocean drift trajectories using massive ensembles and GPUs. In Klöfkorn, R., Keilegavlen, E., Radu, F., and Fuhrmann, J., editors, *Finite Volumes for Complex Applications IX - Methods, Theoretical Aspects, Examples*, pages 715–723, Cham. Springer International Publishing.

Holm, H., Sætra, M., and van Leeuwen, P. (2020c). Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting. *Journal of Computational Physics: X*, 6(0314):100053.

Huang, S., Xiao, S., and Feng, W.-C. (2009). On the energy efficiency of graphics processing units for scientific computing. In *2009 IEEE International Symposium on Parallel & Distributed Processing*, pages 1–8. IEEE.

Kim, C.-H. and Yoon, J.-H. (1996). Modeling of the wind-driven circulation in the Japan Sea using a reduced gravity model. *Journal of Oceanography*, 52:359–373.

Kristensen, N., Røed, L., and Sætra, Ø. (2022). A forecasting and warning system of storm surge events along the Norwegian coast. *Environmental Fluid Mechanics*, pages 1–23.

Kudryavtsev, V., Monzikova, A., Combot, C., Chapron, B., and Reul, N. (2019). A simplified model for the baroclinic and barotropic ocean response to moving tropical cyclones: 2. Model and simulations. *Journal of Geophysical Research: Oceans*, 124(5):3462–3485.

Large, W. and Pond, S. (1981). Open ocean momentum flux measurements in moderate to strong winds. *Journal of Physical Oceanography*, 11:324–336.

Madec, G. and the NEMO team (2008). *NEMO ocean engine*. Note du Pôle de modélisation, Institut Pierre-Simon Laplace (IPSL), France, No 27, ISSN No 1288-1619.

NVIDIA (2023). NVIDIA CUDA C++ programming guide version 12.2.

Peña-Molino, B., Rintoul, S., and Mazloff, M. (2014). Barotropic and baroclinic contributions to along-stream and across-stream transport in the Antarctic Circumpolar Current. *Journal of Geophysical Research: Oceans*, 119(11):8011–8028.

Røed, L. (2012). Documentation of simple ocean models for use in ensemble predictions. Part I: Theory. Technical report, Norwegian Meteorological Institute.

Røed, L. (2019). *Atmospheres and Oceans on Computers*. Springer International Publishing.

Røed, L., Kristensen, N., Hjelmervik, K., and Staalstrøm, A. (2016). A high-resolution, curvilinear ROMS model for the Oslofjord. FjordOs technical report no. 2. Technical report, MET Norway.

Röhrs, J. and Christensen, K. (2015). Drift in the uppermost part of the ocean. *Geophysical Research Letters*, 42(23):10–349.

Röhrs, J., Sutherland, G., Jeans, G., Bedington, M., Sperrevik, A., Dagestad, K.-F., Gusdal, Y., Mauritzen, C., Dale, A., and LaCasce, J. (2023). Surface currents in operational oceanography: Key applications, mechanisms, and methods. *Journal of Operational Oceanography*, 16(1):60–88.

Samelson, R., Chelton, D., and Schlax, M. (2019). The ocean mesoscale regime of the reduced-gravity quasigeostrophic model. *Journal of Physical Oceanography*, 49(10):2469–2498.

Sanders, J. and Kandrot, E. (2010). *CUDA by Example: An Introduction to General-Purpose GPU Programming*. Addison-Wesley Professional.

Shchepetkin, A. and McWilliams, J. (2005). The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9(4):347–404.

Staalstrøm, A., Aas, E., and Liljebladh, B. (2012). Propagation and dissipation of internal tides in the Oslofjord. *Ocean Science*, 8(4):525–543.

Støylen, E. and Fer, I. (2014). Tidally induced internal motion in an Arctic fjord. *Nonlinear Processes in Geophysics*, 21(1):87–100.

Sutherland, G., Soontiens, N., Davidson, F., Smith, G., Bernier, N., Blanken, H., Schillinger, D., Marcotte, G., Röhrs, J., Dagestad, K.-F., et al. (2020). Evaluating the leeway coefficient of ocean drifters using operational marine environmental prediction systems. *Journal of Atmospheric and Oceanic Technology*, 37(11):1943–1954.

Van Leeuwen, P. (2009). Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12):4089–4114.

van Sebille, E., Aliani, S., Law, K., Maximenko, N., Alsina, J., Bagaev, A., Bergmann, M., Chapron, B., Chubarenko, I., Cózar, A., Delandmeter, P., Egger, M., Fox-Kemper, B., Garaba, S. P., Goddijn-Murphy, L., Hardesty, B. D., Hoffman, M. J., Isobe, A., Jongedijk, C. E., Kaandorp, M. L. A., Khatmullina, L., Koelmans, A. A., Kukulka, T., Laufkötter, C., Lebreton, L., Lobelle, D., Maes, C., Martinez-Vicente, V., Maqueda, M. A. M., Poulain-Zarcos, M., Rodríguez, E., Ryan, P. G., Shanks, A. L., Shim, W. J., Suaria, G., Thiel, M., van den Bremer, T. S., and Wichmann, D. (2020). The physical oceanography of the transport of floating marine debris. *Environmental Research Letters*, 15(2):023003.

Wang, J., Ikeda, M., and Saucier, F. (2003). A theoretical, two-layer, reduced-gravity model for descending dense water flow on continental shelves/slopes. *Journal of Geophysical Research: Oceans*, 108(C5).

A Derivation of Reduced-Gravity Model

Starting from the two-layer shallow-water equations, here shown with the momentum in x-direction only for simplicity,

$$\begin{bmatrix} h_1 \\ h_1u_1 \\ h_2 \\ h_2u_2 \end{bmatrix}_t + \begin{bmatrix} h_1u_1 \\ h_1u_1^2 + \frac{1}{2}gh_1^2 \\ h_2u_2 \\ h_2u_2^2 + \frac{1}{2}gh_2^2 \end{bmatrix}_x = \begin{bmatrix} 0 \\ fh_1v_1 \\ 0 \\ -fh_2v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -gh_1(h_2)_x \\ 0 \\ -g\delta h_2(h_1)_x \end{bmatrix} + \begin{bmatrix} 0 \\ gh_1H_x \\ 0 \\ gh_2H_x \end{bmatrix} + \begin{bmatrix} 0 \\ \tau_u - r(u_1-u_2)|u_1-u_2| \\ 0 \\ r_1(u_1-u_2)|u_1-u_2| - r_2u_2|u_2| \end{bmatrix}. \quad \text{(A.1)}$$

The variables are sketched in Figure A.1. Subscripts 1 and 2 denote upper and lower layers, respectively, and total equilibrium depth is $H = H_1 + H_2$. Let η and ζ denote deviations from equilibrium levels at the surface and layer interface, respectively, so that $h_1 = H_1 + \eta - \zeta$ and $h_2 = H_2 + \zeta$. Furthermore, τ_u is wind forcing, and r_1 and r_2 are the friction coefficients accounting for the shear stress between the layers and the bed friction, respectively. Finally, $\delta = \rho_1/\rho_2 < 1$, where ρ is water density.

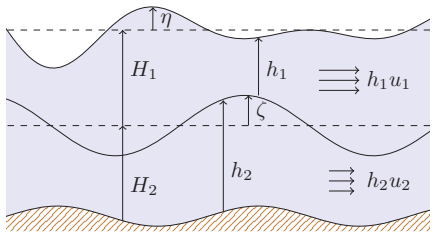


Figure A.1: Two-level shallow-water model with variables and reference levels.

To derive the baroclinic reduced-gravity model we first isolate the baroclinic response by enforcing $\eta \equiv 0$, which requires

$$h_1u_1 = -h_2u_2. \quad \text{(A.2)}$$

Consequently, the mass conservation in both layers reduces to the same equation

$$(-\zeta)_t + (h_1u_1)_x = 0. \quad \text{(A.3)}$$

The resulting system of equations could also be a valid modeling approach, but would require very different numerical methods compared to those used herein.

Next, we assume that the lower layer is much deeper than the upper layer, $h_1 \ll h_2$. Using (A.2) once again, this means that $|u_2| \ll |u_1|$, and we can think of the lower layer as almost stationary. With $u_2, v_2 = 0$, the momentum equation in the lower layer in (A.1) reduces to

$$(\tfrac{1}{2}gh_2^2)_x = -\delta gh_2(h_1)_x + gh_2H_x. \quad \text{(A.4)}$$

Using chain rule, this can be also written as

$$(h_2)_x \approx -\delta(h_1)_x + H_x. \tag{A.5}$$

By looking at only the gravitational terms in the momentum equation for the upper layer, while putting all other terms into F and plugging (A.5) in, we get that

$$\begin{aligned} \left(\tfrac{1}{2}gh_1^2\right)_x &= -gh_1(h_2)_x + gh_1H_x + F, \\ gh_1(h_1)_x &= \delta gh_1(h_1)_x - gh_1H_x + gh_1H_x + F, \\ (1-\delta)gh_1(h_1)_x &= F, \\ (1-\delta)\left(\tfrac{1}{2}gh_1^2\right)_x &= F, \end{aligned} \tag{A.6}$$

where the forcing term from the bathymetry got cancelled out.

Finally, let $g' = (1-\delta)g = \frac{\rho_2-\rho_1}{\rho_2}g$. Note that we then can discard the momentum equation for the lower level in (A.1), which leaves us with

$$\begin{bmatrix} -\zeta \\ h_1u_1 \end{bmatrix}_t + \begin{bmatrix} h_1u_1 \\ h_1u_1^2 + \frac{1}{2}g'h_1^2 \end{bmatrix}_x = \begin{bmatrix} 0 \\ fh_1v_1 \end{bmatrix} + \begin{bmatrix} 0 \\ \tau_u - r_1u_1|u_1| \end{bmatrix}. \tag{A.7}$$

By changing the orientation of ζ, replacing all subscripts $\cdot_1$ with bars $\tilde{\cdot}$, and including the momentum in y-direction, this corresponds to the baroclinic reduced-gravity model in (2.3).

An alternative derivation can be found in Røed (2012, Sec 2.7). There, it is also suggested to use a reduced gravity that varies in space and time. The proposed reduced gravity would then take values in the interval $\left[\frac{g'}{2}, g'\right]$. Seeing the low sensitivity with respect to an even larger uncertainty interval in Section 4, we neglect this suggestion in favor for re-using the code infrastructure of the barotropic model.

B Internal Kelvin Waves

Kelvin waves are coastally trapped through the Coriolis force and limited in spatial extent to within a Rossby radius of deformation from the coastline. In the northern hemisphere, Kelvin waves propagate with the coast to the right. For our applications with reduced-gravity models, it is the internal Kelvin wave and internal Rossby radius that are relevant. A practical example of such internal motion is studied for example in (Støylen and Fer, 2014). The internal Rossby radius is calculated from the reduced-gravity phase speed, $c = \sqrt{g'H}$, with a corresponding Rossby radius of $R = c/f$. We consider an idealized set-up to verify the simulation results by choosing a similar case to that presented in Holm et al. (2020a).

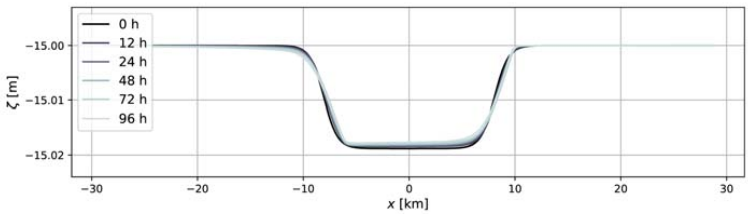
We use a rectangular domain with size 80 km × 120 km with wall boundary conditions at the north and south boundaries, and a periodic boundary in east and west. The initial state is constructed as

$$\begin{aligned} \zeta(x,y,0) &= \zeta_0 \ \exp\left(\frac{-\sqrt{(y-y_0)^2}}{R}\right) F(x), \\ hu(x,y,0) &= c \cdot \operatorname{sign}(y-y_0)\zeta(x,y,0), \qquad hv(x,y,0) = 0, \end{aligned} \tag{B.1}$$

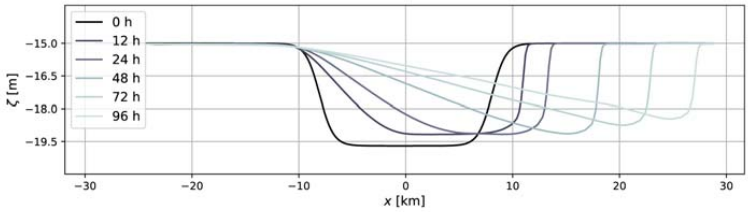
with the shape function

$$F(x) = 1.0 + \tanh\left(\frac{-\sqrt{(x-x_0)^2} + d_F}{R/l_F}\right),$$

where $(x_0, y_0) = (0, -200\,\mathrm{m})$ is a reference point at the center of the domain in x but outside of the southern boundary located at $y = 0$. Furthermore, d_F and l_F are shape parameters for the initial wave where we use the value 2.5 for both, and ζ_0 is a scaling. The constructed wave travels in x-direction along the wall at $y = 0$. The reduced gravity is set to $g' = 0.01\,\frac{\mathrm{m}}{\mathrm{s}^2}$ and the equilibrium depth to $H = -15\,\mathrm{m}$. The resulting



(a) Small wave



(b) Big wave

Figure B.1: Simulated Kelvin waves along the boundary at $y = 0$, shifted along the x-axis according to the theoretic wave speed c. The darkest line shows the initial state of ζ and the direction of motion is towards the right.

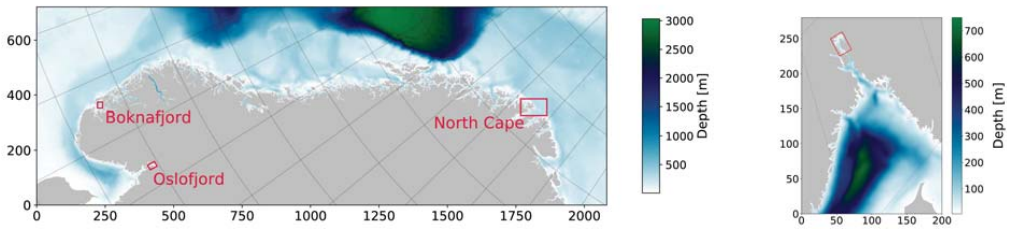
phase speed becomes $c \approx 0.39\,\mathrm{m/s}$, which is in big contrast to barotropic waves that typically have phase speeds in the range of $10 - 100\,\mathrm{m/s}$.

For a uniformly discretized domain with $\Delta x = 200\,\mathrm{m}$ and $\Delta y = 400\,\mathrm{m}$, we run the numerical experiment for a small wave ($\zeta_0 = 0.05$) and a big wave ($\zeta_0 = 2.5$). Figure B.1 shows the evolution of the two Kelvin waves, where we have shifted the solution along the x-axis according to the theoretical phase speed. For the small wave, we see that the Kelvin wave maintains the initial shape well and travels with the phase speed c. This is expected since the non-linear terms in the model become negligible for sufficiently small waves. With a larger amplitude, the non-linearities become clearly visible in the solution, causing the shape of the Kelvin wave to deform towards a shock. This is also expected since the water depth is larger at the top of the wave than on the tail, causing the wave top to travel slightly faster. The wave is nevertheless maintained along the wall boundary, and hence demonstrates that Kelvin waves are well captured in the reduced-gravity model.

C Details on Initial Conditions

Figure C.1 explains the location of the cases considered in previous sections. For the Barents Sea, we use initial conditions from NorFjords at 2019-02-01 10:00. FjordOs provides initial fields for the Oslofjord at 2014-07-06 09:00:00. The Boknafjord simulation are initialized from NorFjords at 2019-11-01 11:00.

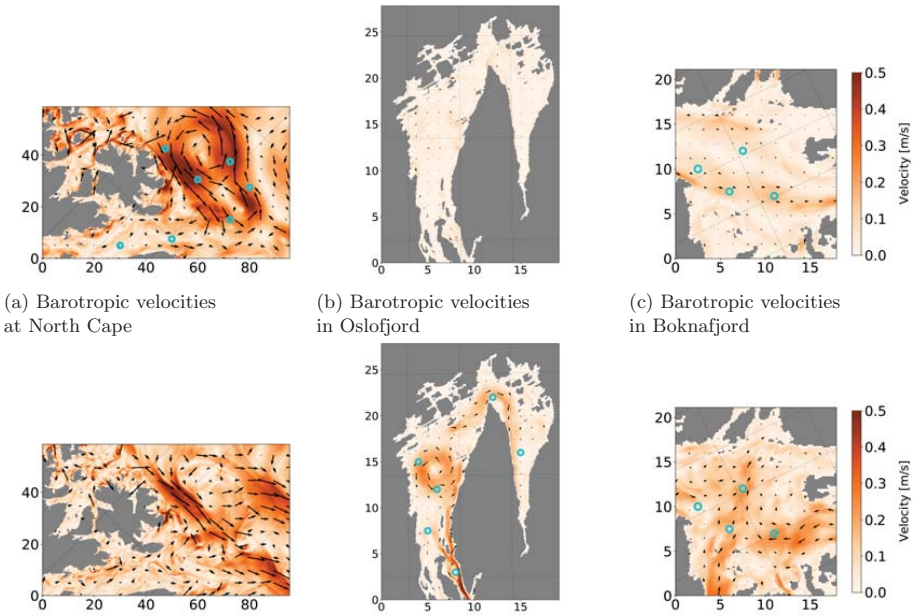
Figure C.2 shows the reference initial conditions for all the three cases in Section 4. The barotropic currents at the North Cape are much stronger than in the other cases. In the Osloford, barotropic currents are remarkably small and usually much smaller than $1\,\frac{\mathrm{cm}}{\mathrm{s}}$. The baroclinic currents in the Oslofjord are significantly stronger in the strait east of Håøya, which is in accordance to the artificial scenario in Section 3.



(a) Areas of the three cases within the Norwegian coastal area

(b) Inner Oslofjord relative to Skagerrak.

Figure C.1: The three cases from Section 4 marked on a map of Norway



(a) Barotropic velocities at North Cape

(b) Barotropic velocities in Oslofjord

(c) Barotropic velocities in Boknafjord

(d) Baroclinic velocities in 25 m-layer at North Cape

(e) Baroclinic velocities in Oslofjord

(f) Baroclinic velocities in Boknafjord

Figure C.2: Reference initial velocities for all cases and turquoise circles marking the initial drifter positions. The numbers along the axes represent km.

Paper III

Comparison of Ensemble-Based Data Assimilation Methods for Sparse Oceanographic Data

Florian Beiser, Håvard Heitlo Holm, Jo Eidsvik

Quarterly Journal of the Royal Meteorological Society

DOI: 10.1002/qj.4637

Comparison of Ensemble-Based Data Assimilation Methods for Sparse Oceanographic Data

Florian Beiser[1,2], Håvard Heitlo Holm[1], and Jo Eidsvik[2]

[1]Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, NTNU, Trondheim, Norway

Abstract

Probabilistic forecasts in oceanographic applications, such as drift trajectory forecasts for search-and-rescue operations, face challenges due to high-dimensional complex models and sparse spatial observations. We discuss localisation strategies for assimilating sparse point observations and compare the implicit equal-weights particle filter and a localised version of the ensemble-transform Kalman filter. First, we verify these methods thoroughly against the analytic Kalman filter solution for a linear advection diffusion model. We then use a non-linear simplified ocean model to do state estimation and drift prediction. The methods are rigorously compared using a wide range of metrics and skill scores. Our findings indicate that both methods succeed in approximating the Kalman filter reference for linear models of moderate dimensions, even for small ensemble sizes. However, in high-dimensional settings with a non-linear model, we discover that the outcomes are significantly influenced by the dependence of the ensemble Kalman filter on relaxation and the particle filter's sensitivity to the chosen model error covariance structure. Upon proper relaxation and localisation parametrisation, the ensemble Kalman filter version outperforms the particle filter in our experiments.

Keywords: Spatio-temporal Statistics, Data Assimilation, Sparse Observations, Oceanographic Applications

1 Introduction

Data assimilation plays an essential role in enhancing the reliability of operational oceanographic and atmospheric forecasts by providing a framework to update and calibrate numerical models using observed data (Evensen, 2009; Asch et al., 2016). There is typically a wide array of available observations, such as satellite imagery, radar measurements, weather stations, and ocean buoys. These observations represent various physical quantities and can exhibit diverse connections to the dynamical model, necessitating substantial pre-processing efforts in operational prediction systems. Furthermore, assimilating the observations and running the complex numerical simulations often demands substantial computational resources.

In addition to regularly updated operational forecasts, there is a crucial need to provide targeted predictions for localized and time-sensitive scenarios, including search-and-rescue operations or sea contamination incidents (Breivik et al., 2013; Röhrs et al., 2018). To gain a better understanding of the local conditions in such situations, it is possible to deploy and gather in-situ observations using drifters. However, due to the fixed schedules for the operational forecasts, such in-situ observations will rarely be processed immediately by the operational data assimilation cycle. Instead, we aim to concurrently assimilate these observations into ensembles of efficient simplified models. This approach complements the traditional operational framework by enabling rapid predictions with quantification of associated uncertainties, possibly without requiring access to supercomputers. This study aims to investigate efficient data assimilation methods tailored for sparse observations for such scenarios.

We limit our scope to point observations, such as information gathered from buoys and drifters. Although these observations provide valuable insight into the ocean state at their respective locations, they are often separated by significant distances, necessitating spatio-temporal modelling to fill the gaps between sparse data points. In this paper, we utilize buoy information for two purposes: i) to constrain an advection diffusion process for particle concentration (Foss et al., 2021), and ii) to constrain drift trajectories in an simplified ocean model (Holm et al., 2020). Case i) involves a linear system in space-time, allowing us to explore the properties of the targeted data assimilation methods for sparse observations by comparing their results to the optimal analytic Kalman Filter (KF) solution. In contrast, case ii) deals with a highly non-linear dynamical model, prompting us to compare the different approaches using various performance metrics through synthetic simulation studies. Our primary motivation for focusing on case ii) is to improve short-term predictions for search-and-rescue missions by leveraging the utilisation of spatially sparse buoy observations.

To effectively model and estimate the uncertainty associated with discretised high-dimensional state variables governed by (non-)linear partial differential equations, we base our work on ensemble prediction and ensemble-based data assimilation. From a statistical perspective, data assimilation methods share the objective of representing a conditional distribution given the available observations. In our case, as we are motivated by search-and-rescue applications, we focus on filtering distributions rather than smoothers that also give improved estimates of past states. The ensemble Kalman filter (EnKF) proposed by Evensen (1994) is widely used in practice, and various numerical adaptions have been developed to address specific practical challenges and problems. Sparse observations pose challenges related to filtering efficiency and quality and our focus centers around the ensemble transform Kalman filter (ETKF, Bishop et al. (2001)) and explores localisation strategies for Kalman filters. In contrast to EnKF and its variants, particle filters (PFs, see, e.g. Van Leeuwen (2009) or Chopin and Papaspiliopoulos (2020)) are appealing as they do not rely on assumptions of a linear model and Gaussian probability distributions, at least in their most basic forms. However, PFs are less commonly employed in high-dimensional real-world applications due to the issue of degeneracy. Holm et al. (2020) demonstrated a modern PF approach based on the implicit equal-weight particle filter (IEWPF) proposed by Zhu et al. (2016), which shows promising forecasting results for drift trajectories.

In this paper, we systematically compare the statistical properties and performance of two ensemble-based data assimilation methods for sparse observations in practical oceanographic applications. We compare the ETKF with localisation for sparse observations against the state-of-the-art particle filter IEWPF. The comparison involves appraising their ability to reproduce analytic solutions for the linear model in the first case by assessing a range of statistical performance metrics and evaluating forecast skills in the non-linear case. Furthermore, we provide comprehensive discussions on the localisation techniques applied to the ETKF and the inherent localisation mechanism within the IEWPF.

The remainder of this article is organised as follows: Section 2 explicates state-of-the-art ensemble-based data assimilation techniques and reviews localisation strategies in the context of sparse observations. In Section 3, we use the dynamical model based on the advection diffusion equation to verify the relevant ensemble-based filtering methods against an analytical solution. Section 4 outlines the non-linear simplified ocean model for drift trajectory prediction, and we compare the performance of the data assimilation methods in this context. Finally, closing remarks are in Section 5.

2 The Data Assimilation Problem and Ensemble-based Filtering

Spatio-temporal quantities are denoted by $\boldsymbol{x}(t, \boldsymbol{s})$, for time $t > 0$ and location $\boldsymbol{s}$ where we restrict ourselves to $\boldsymbol{s} \in \mathbb{R}^2$. Upon discretisation of the spatio-temporal domain of interest, the locations are represented at grid nodes of spatial locations $(\boldsymbol{s}_i)_{i=1}^{N_s}$ and time steps t^n, $n = 1, \ldots, N_T$, where we will work with equidistant time steps without loss of generality. The *state vector* at time t^n is denoted $\boldsymbol{x}^n \in \mathbb{R}^{N_X}$ and can hold more than one physical variable per location if necessary ($N_X \geq N_s$). In oceanographic applications, the dimension N_X is usually very high due to large domains and several physical variable.

The numerical model is embraced in the *model operator* $\mathcal{M}$. It propagates the state vector from the

previous time step t^{n-1} to the current t^n, defining the so-called forecast state. The model usually describes the physics of the ocean. To account for uncertainty coming from external factors, unknown model parameters and non-modelled physics, a Gaussian model error $\boldsymbol{\nu}^n \sim \mathcal{N}(0, \boldsymbol{Q})$ is added every time step. It is assumed that $\boldsymbol{Q}$ is known, and that the error terms are uncorrelated in time. Starting from initial state $\boldsymbol{x}^0$, the model evolves as

$$\boldsymbol{x}^n = \mathcal{M}\boldsymbol{x}^{n-1} + \boldsymbol{\nu}^n, \quad n = 1, \dots \ . \tag{2.1}$$

The oceanographic state $\boldsymbol{x}^n$ is often only partially observed and sometimes even indirectly. An *observation* is extracted from the true state vector by the measurement operator $\boldsymbol{H}$. Due to the nature of the problems within this work, we impose the restriction that this operator is linear, and that $\boldsymbol{H}$ only extracts variables directly from a subset of spatial locations, meaning that the matrix consists of at most one 1 entry per row and otherwise 0 entries. The observation locations could change every time step, but for the sake of eased reading we ignore this in our notation. Measurement and representation inaccuracies are represented by the addition of a zero-mean Gaussian error $\boldsymbol{\epsilon}^n \sim \mathcal{N}(0, \boldsymbol{R})$ with known covariance matrix $\boldsymbol{R}$ and no correlations in time. We model observations $\boldsymbol{y}^n \in \mathbb{R}^{N_Y}$ by

$$\boldsymbol{y}^n = \boldsymbol{H}\boldsymbol{x}^n + \boldsymbol{\epsilon}^n, \tag{2.2}$$

and assume that data comes from this observation model employed on the unknown true state. The covariance matrix $\boldsymbol{R}$ is assumed to be diagonal, representing conditional independence between the observations given the state variables. A characteristic of many oceanographic applications, and a key assumption in our setting, is that in-situ observations are spatially sparse and of low dimension compared to the high-dimensional state vector, i.e.,

$$N_Y \ll N_X. \tag{2.3}$$

Sequential *data assimilation* refers to the workflow of sequentially updating the probability density of the state variables as more data gets available. Often, this results in reduced uncertainty, especially in the neighbourhood of the observed variables. This kind of data assimilation is formalised in Bayes' rule. Using the state's density conditioned on all previous observations $p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1})$ as the *forecast* (prediction or *prior*), this rule is used to assimilate the new observation $\boldsymbol{y}^n$ and thus provides the *analysis* (filtering or *posterior*) density $p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n})$ of the state. With the independence assumption for the observation noise terms, this formalism can be applied recursively as

$$p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n}) \propto p(\boldsymbol{y}^n|\boldsymbol{x}^n)p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1}), \quad n = 1, 2, \dots, \tag{2.4}$$

starting with only prior information at the first time step.

Kalman filtering In the case of a linear model $\mathcal{M} = \boldsymbol{M}$ and an initial Gaussian distribution for the state $\boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{\mu}^0, \boldsymbol{\Sigma}^0)$, all forecast and analysis distributions remain Gaussian. The data assimilation problem is then Gauss-linear and Bayes' formula (2.4) takes closed form solutions for the mean vectors and covariance matrices. Assuming that we have an analysis (superscript a) distribution characterised by the mean $\boldsymbol{\mu}^{n-1,a}$ and covariance matrix $\boldsymbol{\Sigma}^{n-1,a}$, the forecast (superscript f) distribution $\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1} \sim \mathcal{N}(\boldsymbol{\mu}^{n,f}, \boldsymbol{\Sigma}^{n,f})$ is obtained by evolving the first and second moments from the previous time step to the next by

$$\boldsymbol{\mu}^{n,f} = \boldsymbol{M}\boldsymbol{\mu}^{n-1,a} \tag{2.5a}$$

$$\boldsymbol{\Sigma}^{n,f} = \boldsymbol{M}\boldsymbol{\Sigma}^{n-1,a}\boldsymbol{M}^\top + \boldsymbol{Q}. \tag{2.5b}$$

The analysis distribution $\boldsymbol{x}^n|\boldsymbol{y}^{1:n} \sim \mathcal{N}(\boldsymbol{\mu}^{n,a}, \boldsymbol{\Sigma}^{n,a})$ is achieved by assimilating the latest observation $\boldsymbol{y}^n$ via Bayes' rule for the given Gaussian model, resulting in

$$\boldsymbol{\mu}^{n,a} = \boldsymbol{\mu}^{n,f} + \boldsymbol{K}(\boldsymbol{y}^n - \boldsymbol{H}\boldsymbol{\mu}^{n,f}) \tag{2.6a}$$

$$\boldsymbol{\Sigma}^{n,a} = \boldsymbol{\Sigma}^{n,f} - \boldsymbol{K}\boldsymbol{\Sigma}^{n,f}\boldsymbol{K}^\top. \tag{2.6b}$$

Here, $\boldsymbol{K} = \boldsymbol{\Sigma}^{n,f}\boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{\Sigma}^{n,f}\boldsymbol{H}^\top + \boldsymbol{R})^{-1}$ is the Kalman gain, which in eq. (2.6a) maps the so-called innovation, $\boldsymbol{y}^n - \boldsymbol{H}\boldsymbol{\mu}^{n,f}$, to the state space using the state covariance matrix and the observation model. From the numerical perspective, note that the Kalman filter requires storage and propagation of the size $N_X \times N_X$ covariance matrix, which can be infeasible for high-dimensional systems.

Ensemble-based data assimilation In oceanographic applications the linearity assumptions of the Kalman filter are rarely met, and linearised approaches can suffer from divergence challenges. There is hence a need for more flexible methods, and ensemble-based approaches have been employed as a computationally feasible method to represent statistical solutions of non-linear systems, even for large N_X.

Therein, the continuous distribution of the state variable is approximated by an ensemble of realisations $(\boldsymbol{x}_e^n)_{e=1}^{N_e}$ and potentially by corresponding weights $(w_e)_{e=1}^{N_e}$. Following the Monte Carlo idea, the marginal distribution of $\boldsymbol{x}^n$ at time t^n becomes

$$p(\boldsymbol{x}^n) \approx \sum_{e=1}^{N_e} w_e^n \delta\left(\boldsymbol{x}^n - \boldsymbol{x}_e^n\right), \tag{2.7}$$

where δ is the Dirac delta function and $\sum_{e=1}^{N_e} w_e^n = 1$.

In the statistical literature (see, e.g. Asch et al. (2016) and Vetra-Carvalho et al. (2018)), there are two popular groups of methods for ensemble-based data assimilation, whose foundations and latest variants for the aforementioned problems is outlined in the next subsections.

2.1 Particle filters in oceanographic applications

PFs are ensemble-based methods for solving the data assimilation problem using the Monte Carlo approach. In their simplest form, starting from a weighted ensemble approximation for $p(\boldsymbol{x}^0)$ or $p(\boldsymbol{x}^{n-1}|\boldsymbol{y}^{1:n-1})$ in the form of eq. (2.7), the forecast distribution $p(\boldsymbol{x}^n|\boldsymbol{y}^{1:n-1})$ can be approximated by propagating each ensemble member $\boldsymbol{x}_e^{n-1}$ individually by the model eq. (2.1) to obtain $\boldsymbol{x}_e^n$. Plugging this into Bayes formula (2.4), the weights are updated as

$$w_e^n \propto p(\boldsymbol{y}^n|\boldsymbol{x}_e^n)p(\boldsymbol{x}_e^n|\boldsymbol{x}_e^{n-1})w_e^{n-1}. \tag{2.8}$$

Since ensemble members that have weights very close to zero do not contribute to the posterior probability distribution, it is common to combine eq. (2.8) with a discrete resampling of the ensemble members based on their weights (see, e.g. Van Leeuwen (2009); Chopin and Papaspiliopoulos (2020) for reviews of resampling schemes). In practice, this means that we discard ensemble members with low weights and duplicate those with higher weights, thus ensuring that computational resources are used to describe the non-negligible part of the probability distribution. In high-dimensional applications, however, these basic PFs are prone to degenerate, i.e. all but one ensemble member get a weight close to zero, leading to loss of statistical properties (Snyder et al., 2008).

Among other concepts, one way to counteract such degeneracy is to sample from a proposal density q_e instead of evolving the ensemble directly according to $p(\boldsymbol{x}_e^n|\boldsymbol{x}_e^{n-1})$ (van Leeuwen et al., 2019). The proposal density can be conditioned on the latest observation $\boldsymbol{y}^n$ and the previous state $\boldsymbol{x}_e^{n-1}$ for all ensemble members $e = 1, \ldots, N_e$. The weights are then modified to

$$w_e^{n,*} = \frac{w_e^n}{q_e(\boldsymbol{x}_e^n|\boldsymbol{x}_{1:N_e}^{n-1}, \boldsymbol{y}^n)}.$$

The variance in the weights can be reduced in this way, and the minimal variance is achieved by $q_e(\boldsymbol{x}^n|\boldsymbol{x}_{1:N_e}^{n-1}, \boldsymbol{y}^n) = p(\boldsymbol{x}^n|\boldsymbol{x}_e^{n-1}, \boldsymbol{y}^n)$ as described in Doucet et al. (2000) and often referred to as the optimal proposal. Under the assumption of Gaussian errors and linear observation operator, which applies to our case, this proposal is a Gaussian distribution $\mathcal{N}\left(\boldsymbol{x}_e^{n,\mathrm{opt}}, \boldsymbol{P}\right)$ with

$$\boldsymbol{x}_e^{n,\mathrm{opt}} = \mathcal{M}\boldsymbol{x}_e^{n-1} + \boldsymbol{Q}\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1}\left(\boldsymbol{y}^n - \boldsymbol{H}\mathcal{M}\boldsymbol{x}_e^{n-1}\right) \tag{2.9a}$$

$$\boldsymbol{P} = \boldsymbol{Q} - \boldsymbol{Q}\boldsymbol{H}^\top\left(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R}\right)^{-1}\boldsymbol{H}\boldsymbol{Q}. \tag{2.9b}$$

Note that the expression for $\boldsymbol{x}_e^{n,\text{opt}}$ is similar to the KF update in eq. (2.6a), but it uses the covariance structure from the model error $\boldsymbol{Q}$ instead of the forecast $\boldsymbol{\Sigma}^{n,f}$. Even this formulation of the PF will degenerate for high-dimensional systems, see, e.g. Morzfeld et al. (2017).

The optimal proposal density filter can be modified such that all posterior ensemble members obtain a certain target weight w_{target}^n, and thus avoiding ensemble collapse and the need of resampling. Instead of drawing realisations from the proposal distribution directly, the IEWPF first samples $\boldsymbol{\xi}_e$ and $\boldsymbol{\zeta}_e$ from $\mathcal{N}(0, \boldsymbol{I}_{N_X})$ and next implicitly transforms the samples to a target distribution. This filter, introduced by Zhu et al. (2016) and modified by Skauvold et al. (2019), utilises a version of the optimal proposal density where $\boldsymbol{\xi}_e$ and $\boldsymbol{\zeta}_e$ are constructed to be perpendicular and scaled according to factors $\alpha_e^{1/2}$ for $e = 1, ..., N_e$ and $\beta^{1/2}$, respectively, before being transformed by $\boldsymbol{P}$ according to

$$\boldsymbol{x}_e^n = \boldsymbol{x}_e^{n,\text{opt}} + \boldsymbol{P}^{1/2}\left(\alpha_e^{1/2}\boldsymbol{\xi}_e + \beta^{1/2}\boldsymbol{\zeta}_e\right). \tag{2.10}$$

Here, the constant β is a tuning parameter that influences the statistical quality of the results, whereas the α_e values are calculated implicitly for each ensemble member to ensure equal weights. To get an intuition of the idea behind IEWPF, we observe that eq. (2.9a) develops the state using $\mathcal{M}$ but with no model error. These unperturbed states are then used to get $\boldsymbol{x}_e^{n,\text{opt}}$ by assimilating the observations. We then perturb $\boldsymbol{x}_e^{n,\text{opt}}$ in (2.10), but instead of using the model error, which is sampled from $\mathcal{N}(0, \boldsymbol{Q})$, we sample perturbations from $\mathcal{N}(0, \alpha_e\boldsymbol{P})$ and $\mathcal{N}(0, \beta\boldsymbol{P})$ in such a way that we counteract the change of weights in the ensemble. Note that $\boldsymbol{P}$ is constructed from $\boldsymbol{Q}$ as seen in eq. (2.9b), so that these perturbations can be thought of as targeted sampling of the model error.

The tuning parameter β needs to be selected with care. A small value gives small spread of the ensemble that likely underestimates the variability, whereas a bigger β increases the spread. Holm et al. (2020, Appendix A) derived lower and upper bounds for this tuning parameter. In the subsequent experiments, we tune β manually, mainly by calibration of coverage probabilities as suggested in Skauvold et al. (2019). By experience it seems that β values around 0.5 are a good start. Albeit the choice of β is independent of the ensemble size, it is influenced by the dynamics of the problem. Hence, one can find a suitable choice of β for a specific kind of scenario and then keep it fixed in future experiments with similar characteristics.

It should be noted that in contrast to general PFs, the IEWPF requires that the model error is additive and from a Gaussian distribution. Furthermore, there are no guarantees on how the IEWPF performs, even when the ensemble size goes to infinity. Still, the performance tends to be very good in large-size systems. With $\beta = 0$, the implicit transform has a gap that leads to asymptotic bias (Skauvold et al., 2019), but this seems to be adjusted reasonably well by the second part having $\beta > 0$. The IEWPF has recently been shown to be applicable and efficient for assimilating point-based observations into a simplified ocean model based on the shallow water equations (Holm et al., 2020). Herein, this method represents a state-of-the-art PF and is investigated more thoroughly.

2.2 Ensemble Kalman filters in oceanographic applications

The EnKF (Evensen, 1994, 2009) is an ensemble-based version of the KF, given in eqs. (2.5) and (2.6). Originally presented as a data assimilation method for non-linear systems, it also solves the problem of having to store and propagate the $N_X \times N_X$ state covariance matrix $\boldsymbol{\Sigma}$.

In the ensemble representation from eq. (2.7), all weights are kept equal, and the state of each ensemble member is propagated by the model in eq. (2.1). The forecast state covariance is estimated from the ensemble, as

$$\widehat{\boldsymbol{\Sigma}}^{n,f} = \frac{1}{N_e - 1}\sum_{e=1}^{N_e}\left(\boldsymbol{x}_e^{n,f} - \overline{\boldsymbol{x}}^{n,f}\right)\left(\boldsymbol{x}_e^{n,f} - \overline{\boldsymbol{x}}^{n,f}\right)^\top, \tag{2.11}$$

where $\overline{\boldsymbol{x}}^{n,f}$ denotes the ensemble mean. The ensemble members are then updated along the same linear projection

$$\boldsymbol{x}_e^{n,a} = \boldsymbol{x}_e^{n,f} + \widehat{\boldsymbol{K}}\left(\boldsymbol{y}^n - \boldsymbol{H}\boldsymbol{x}_e^{n,f} - \boldsymbol{\epsilon}_e^n\right), \tag{2.12}$$

where the Kalman gain becomes $\widehat{\boldsymbol{K}} = \widehat{\boldsymbol{\Sigma}}^{n,f}\boldsymbol{H}^\top(\boldsymbol{H}\widehat{\boldsymbol{\Sigma}}^{n,f}\boldsymbol{H}^\top + \boldsymbol{R})^{-1}$. In eq. (2.12), the perturbation $\boldsymbol{\epsilon}_e^n \sim \mathcal{N}(0, \boldsymbol{R}^n)$ is added to adjust the variance in the solution ensemble, motivated by exact sampling in the linear Gaussian situation. This solution is therefore termed the stochastic EnKF (SEnKF, see (Burgers et al., 1998; Houtekamer and Mitchell, 1998; van Leeuwen, 2020)). The classical SEnKF in eq. (2.12) requires that we obtain and store the relevant covariances from the ensemble, and then factorize matrices to solve a linear system. For high-dimensional applications, this quickly becomes expensive, and it is therefore common to circumvent the covariance assembling (Evensen, 2003) or to use so-called deterministic square-root formulations instead (Whitaker and Hamill, 2002).

To avoid working in the state space, the ensemble transform Kalman filter (ETKF, Bishop et al. (2001)) reformulates eq. (2.12) via linear algebraic identities into a particular example of a deterministic square-root filter, which works in ensemble dimensions instead. Mathematically, let $\boldsymbol{X}^{n,f} = \left[\boldsymbol{x}_1^{n,f}, \ldots, \boldsymbol{x}_{N_e}^{n,f}\right]$ be the matrix of prior ensemble states, and let $\overline{\boldsymbol{X}}^{n,f}$ be a $N_X \times N_e$ matrix where all columns are $\overline{\boldsymbol{x}}^{n,f}$. The ETKF then works on the state perturbation matrix $\boldsymbol{X}_{\text{pert}}^{n,f} = \boldsymbol{X}^{n,f} - \overline{\boldsymbol{X}}^{n,f}$, and calculates the mean of the analysis ensemble

$$\overline{\boldsymbol{X}}^{n,a} = \overline{\boldsymbol{X}}^{n,f} + \boldsymbol{X}_{\text{pert}}^{n,f}\boldsymbol{A}(\boldsymbol{H}\boldsymbol{X}_{\text{pert}}^{n,f})^\top\boldsymbol{R}^{-1}\left(\boldsymbol{y}^n - \boldsymbol{H}\overline{\boldsymbol{x}}^{n,f}\right), \tag{2.13}$$

where

$$\boldsymbol{A} = \left((N_e - 1)\boldsymbol{I}_{N_e} + (\boldsymbol{H}\boldsymbol{X}_{\text{pert}}^{n,f})^\top\boldsymbol{R}^{-1}\boldsymbol{H}\boldsymbol{X}_{\text{pert}}^{n,f}\right)^{-1} \tag{2.14}$$

plays the role of the analysis covariance matrix. The ensemble members are then spread around $\overline{\boldsymbol{x}}^{n,a}$ according to

$$\boldsymbol{X}^{n,a} = \overline{\boldsymbol{X}}^{n,a} + \boldsymbol{X}_{\text{pert}}^{n,f}\left((N_e - 1)\boldsymbol{A}\right)^{1/2}, \tag{2.15}$$

where we use a singular-value decomposition to find the square-root of $\boldsymbol{A}$. The properties of the ETKF remain the same as for the EnKF and we refer to Li (2007) for further details on the transform.

The derivation of these methods assume a linear model, and asymptotic convergence results for increased ensemble size cannot be proved for non-linear cases. Still, the EnKF and its variants have been prevalent and successfully used in oceanographic applications (see, e.g. Carrassi et al. (2018)).

The error covariance matrix in this kind of filters is estimated from the ensemble and can lead to systematic underestimation. Typical approaches to counteract this are inflation or localisation. Anderson and Anderson (1999) introduced *covariance inflation* by a multiplicative factor to keep more variability in the ensemble, where several suggestions for the determination of an adaptive factor exist in literature, see, e.g. Desroziers et al. (2006); Anderson (2009); Sætrom and Omre (2013); Raanes et al. (2019). Similarly, additive inflation was presented by (Ott et al., 2004). However, Li et al. (2009) also point out that covariance inflation may not work appropriately in large complex models. Hence, we will mainly concentrate on localisation.

2.3 Sparse observations

The focus of this paper is on assimilating spatially sparse point observations. This naturally suggests to look closer on *localisation* in the filters. Although localisation is important for general applications, the sparse observations scenario considered here motivates one to study specific methods with good assimilation quality and algorithmic efficiency.

Localisation and sparse observation handling in the IEWPF The need for localisation in EnKF-based schemes arises from the spurious correlations introduced by the term $\widehat{\boldsymbol{\Sigma}}^{n,f}\boldsymbol{H}^\top$, which represents the estimated covariance terms between all state variables and all observations in eq. (2.12). As pointed out in Section 2.1, the optimal proposal distribution in eq. (2.9a) updates the state vector with a similar expression, but it uses the correlations in the model error, $\boldsymbol{Q}\boldsymbol{H}^\top$, rather than the empirical $\widehat{\boldsymbol{\Sigma}}^{n,f}\boldsymbol{H}^\top$. This means that the optimal proposal filter by design does not lead to spurious correlations. Still, the structure of $\boldsymbol{Q}$ of course influences the distribution, and in particular a local structure in $\boldsymbol{Q}$ that do not overlap between observation sites, entails updates in eq. (2.9a) that are local as well.

To show that the IEWPF updates in eqs. (2.9a) and (2.10) are local if $\boldsymbol{Q}$ is local, we consider the pattern of non-zero values in the matrix operations in the two equations. First, let us use r_Q to denote the radius in terms of number of grid cells that information is spread through the model error covariance matrix. This means that $\boldsymbol{Q}^{1/2}$ contains at most $(2r_Q+1)^2$ non-zero elements, whereas $\boldsymbol{Q} = \boldsymbol{Q}^{\top/2}\boldsymbol{Q}^{1/2}$ spreads the information twice as far, thus having at most $(4r_Q+1)^2$ non-zero elements. Second, we observe that eq. (2.9b) can be rewritten as

$$\boldsymbol{P} = \boldsymbol{Q}^{\top/2}\Big(\boldsymbol{I}_{N_X} - \underbrace{\boldsymbol{Q}^{1/2}\boldsymbol{H}^\top(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R})^{-1}\boldsymbol{H}\boldsymbol{Q}^{\top/2}}_{=:\boldsymbol{S}}\Big)\boldsymbol{Q}^{1/2}, \tag{2.16}$$

where we for convenience use $\boldsymbol{S}$ to represent the longest matrix expression. This means that $\boldsymbol{P}^{1/2} = (\boldsymbol{I}_{N_X} - \boldsymbol{S})^{1/2}\boldsymbol{Q}^{1/2}$, which is what we need in eq. (2.10).

In Table 1, we give upper bounds on the number of non-zero elements when stepping through the matrix operations in eqs. (2.9a) and (2.16). In the rightmost column, we assume that observations are sufficiently sparse so that both $\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top$ and $\boldsymbol{R}$ are diagonal, whereas we make no such assumption in the second column from the right. We do however assume that $\boldsymbol{H}$ maps a subset of the state variables directly to observational space. If we now consider a single observation, meaning $N_Y = 1$ and scalar $(\boldsymbol{H}\boldsymbol{Q}\boldsymbol{H}^\top + \boldsymbol{R})^{-1}$, we see from Table 1 that the innovation in eq. (2.9a) is spread in the neighbourhood of the observation location within a radius of $2r_Q$ grid cells. Thus, $\boldsymbol{x}_e^{n,\text{opt}}$ differs from $\mathcal{M}\boldsymbol{x}_e^{n-1}$ only locally around the observation.

To see the resulting non-zero pattern for $\boldsymbol{S}$ in eq. (2.16), we realize that $\boldsymbol{H}\boldsymbol{Q}^{\top/2}$ consists of a single row from $Q^{\top/2}$. Furthermore, we have that the leftmost term $\boldsymbol{Q}^{1/2}\boldsymbol{H}^\top$ is the same as $(\boldsymbol{H}\boldsymbol{Q}^{\top/2})^\top$. Consequently, $\boldsymbol{S}$ is simply the (scaled) outer product of the column from $\boldsymbol{Q}^{1/2}$ corresponding to the observation, and therefore has the local correlation pattern in both its rows and columns, but zero in all rows and columns for which the observed state variable is not correlated with through $\boldsymbol{Q}^{1/2}$. This then means that $(\boldsymbol{I}_{N_X} - \boldsymbol{S})$ differs from the identity only locally to the observed state variable as well, which finally means that $\boldsymbol{P}^{1/2} = (\boldsymbol{I}_{N_X} - \boldsymbol{S})^{1/2}\boldsymbol{Q}^{1/2}$ differs from $\boldsymbol{Q}^{1/2}$ only locally to the observations as well.

It should be noted though, that the values of α_e and β depend on the innovation obtained from all observations in the domain. These parameters are therefore global parameters, but since they are scalars, they do not contribute to any change in the local correlation structures. Related discussions on how to utilize local covariance structures and sparse observations for efficient implementations of the IEWPF can be found in Holm (2020).

In addition to demonstrating the built-in localisation in the IEWPF, Table 1 shows how sensitive the IEWPF is to the structure of the model error covariance matrix $\boldsymbol{Q}$ for spreading observed information in the state space. In the extreme case of a diagonal $\boldsymbol{Q}$ (meaning $r_Q = 0$) all matrix operations will contain exactly N_Y non-zeros, and only the observed state variables will be affected by the data assimilation.

Localisation in the EnKF In a statistical sense, the spurious correlations in the EnKF are due to a poor Monte Carlo approximation of the true covariance matrix, see, e.g. Houtekamer and Zhang (2016). In the spatio-temporal physical model, information propagates at finite speed and long-distance correlations are unlikely to be significant. Prevailing techniques to counteract these artefacts are covariance or observation localisation as they are outlined in Sakov and Bertino (2011). Both of these exploit the physical distance between two points in space to reduce information propagation effects, and this has been demonstrated to work well in practice, see e.g. Soares et al. (2021). For many oceanographic applications, it is important that the geostrophic imbalance introduced by the localisation in the EnKF does not outweigh the natural imbalance - Greybush et al. (2011) provide a discussion and representative experiments to this issue.

For the ETKF, Ott et al. (2004) introduce an efficient localisation scheme and this is referred to as LETKF. Further developed implementations using parallelisation and observation batching exist, see e.g.

Table 1: Number of non-zero elements in the matrix operations used in IEWPF

Eq.	Matrix operations	Dims	# non-zeros for any $(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}$	# non-zeros for diag $(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}$
	$\boldsymbol{Q}^{1/2}$	$N_X \times N_X$	$\leq (2r_Q+1)^2 N_X$	$\leftarrow$
	$\boldsymbol{Q}$	$N_X \times N_X$	$\leq (4r_Q+1)^2 N_X$	$\leftarrow$
	$\boldsymbol{H}$	$N_Y \times N_X$	N_Y	$\leftarrow$
	$(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}$	$N_Y \times N_Y$	$\leq {N_Y}^2$	N_Y
	$\boldsymbol{H}^\top(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}$	$N_X \times N_Y$	$\leq {N_Y}^2$	N_Y
(2.9a)	$\boldsymbol{QH}^\top(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}$	$N_X \times N_Y$	$\leq (4r_Q+1)^2 {N_Y}^2$	$\leq (4r_Q+1)^2 N_Y$
	$\boldsymbol{HQ}^{\top/2}$	$N_Y \times N_X$	$\leq (2r_Q+1)^2 N_Y$	$\leftarrow$
	$(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}\boldsymbol{HQ}^{\top/2}$	$N_Y \times N_X$	$\leq (2r_Q+1)^2 {N_Y}^2$	$\leq (2r_Q+1)^2 N_Y$
	$\boldsymbol{H}^\top(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}\boldsymbol{HQ}^{\top/2}$	$N_X \times N_X$	$\leq (2r_Q+1)^2 {N_Y}^2$	$\leq (2r_Q+1)^2 N_Y$
$\boldsymbol{S}$ in (2.16)	$\boldsymbol{Q}^{1/2}\boldsymbol{H}^\top(\boldsymbol{HQH}^\top + \boldsymbol{R})^{-1}\boldsymbol{HQ}^{\top/2}$	$N_X \times N_X$	$\leq (2r_Q+1)^4 {N_Y}^2$	$\leq (2r_Q+1)^4 N_Y$

Hunt et al. (2007), and the LETKF variants are popular in numerical weather prediction, see e.g., Szunyogh et al. (2007), where for example global satellite data is common. In the LETKF, one loops over the state locations or sets of state locations in a batch area. Doing so, one updates state variables by means of the ETKF using only a specified set of local observations per batch.

Sparse observation handling in the ETKF In an oceanographic scenario with observations at only a few locations, we prefer to use covariance localisation (Houtekamer and Mitchell, 2001) to achieve computational control as we explain further down. One then defines local domains around each observation site only, where the size N_X^{loc} of a local area is significantly smaller than the full state space. For the choice of the radius of the resulting local domains, several approaches exist, see e.g. Kirchgessner et al. (2014), but we advocate using model-informed radii such as the model error range. We next assume that observations with non-overlapping local areas have negligible correlation, and they can be updated separately. Due to the motivating case with sparse observations, we expect to have few overlapping areas. The reduced dimension of the local area compared to the full state vector will make computations more efficient.

Still, with non-linear dynamical models, it is sometimes difficult to predict the possibly undesired effects of local approximations. Using sequential data integration, one can run through the data in multiple assimilation steps and in doing so one properly accounts for the correlations. In cases of overlapping local observation areas, we therefore recommend splitting the observations into observational batches $\mathcal{B}_b, b = 1, \ldots, B$ of assumed uncorrelated observations for serial processing as originally introduced in Houtekamer and Mitchell (1998). In our context, the batches at each step are constructed from far-apart observations. The local areas and the sequential processing give good control for handling correlations from sparse spatial observations. Nerger (2015) discusses how interactions of localisation and serial observation processing could destabilise the filter, but with reasonably set local areas and minimal correlation within a batch, this effect seems to be minuscule.

Figure 2.1(left) illustrates the definition of local areas around depicted observation sites. The figure further indicates the splitting into batches (middle and right display), with observations sites within a batch being sufficiently far apart from each other.

For the covariance localisation, we consider weighting vectors $\boldsymbol{w}_b$ and assume a tapering that assures $\boldsymbol{w}_b = 1$ at an observation site and $\boldsymbol{w}_b = 0$ outside the local areas. While veering away from an observation site, $\boldsymbol{w}$ should transit decreasingly monotone and smoothly from one to zero. An example of a kernel fulfilling those requirements locally in continuous space is the Gaspari-Cohn function (GC) introduced in Gaspari and Cohn (1999), which enjoys popularity in EnKF-localisation. We let $\boldsymbol{w}_{\text{loc},j} \in \mathbb{R}^{N_X^{\text{loc}}}$ be a properly scaled discretisation of the GC kernel around observation j, such that its support matches the radius of the

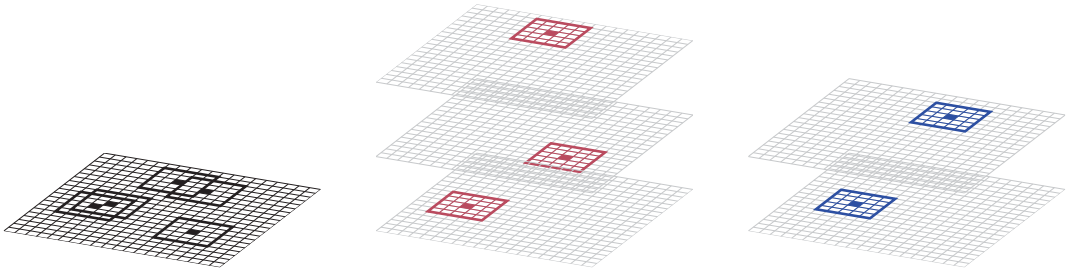


Figure 2.1: Schematic decomposition of the physical space (left panel) into local areas $N_X^{\mathrm{loc}} \ll N_X$ around observations, indicated by a filled grid cell, and the separation into two batches of uncorrelated observations (middle and right panel)

local area. With the same notation, the weighting is composed as

$$\boldsymbol{w}_b = \sum_{j \in \mathcal{B}_b} \boldsymbol{w}_{\mathrm{loc},j}. \tag{2.17}$$

For the observation batch from Figure 2.1, the weighting vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are illustrated in Figure 2.2 together with the contributions from $\boldsymbol{w}_{\mathrm{loc},1}, \ldots, \boldsymbol{w}_{\mathrm{loc},5}$.

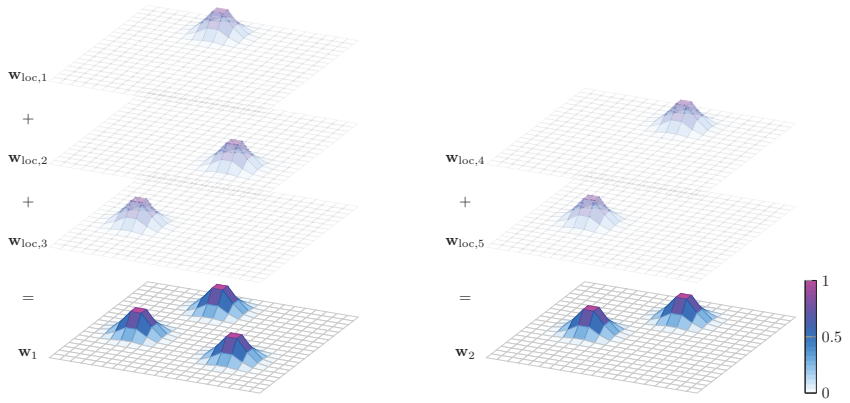


Figure 2.2: Schematic construction of the weighting $\boldsymbol{w}_b$ using the $\boldsymbol{w}_{\mathrm{loc},j}$ within the groups: For batch 1 and 2 from Figure 2.1, the weighting vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are built from the Gaspari-Cohn kernel around each observation with in the two groups. At the corresponding center, each $\boldsymbol{w}_{\mathrm{loc},j}$ equals one and decays towards the boundary of the local domains. By definition the supports are non-overlapping and the contributions are added up.

Within the recursion of batches $\mathcal{B}_b$, the local analysis states $\boldsymbol{x}^{n,a}(j) \in \mathbb{R}^{N_X^{\mathrm{loc}}}$ are calculated independently in each local area around the observation sites $j \in \mathcal{B}_b$, whereby the computational overhead in presence of sparse observations is implicitly controlled as the assimilation scheme operates in much smaller dimensions than N_X. Note that the processing of a batch can hence influence the ensemble that is used as the next forecast. The analyses only take values in their respective small local regions, but to avoid cumbersome notation for transformations, we abuse the same notation for their extension to the full state size $\boldsymbol{x}^{n,a}(j) \in \mathbb{R}^{N_X}$.

Because the ETKF constructs neither the covariance matrix nor the Kalman gain, we do not incorporate the weights directly into the Kalman update, but rather calculate the ETKF analysis state by eq. (2.15) and weight it with the forecast afterwards. (Recall that $\boldsymbol{x}^{n,a}(j)$ are the columns of $\boldsymbol{X}^{n,a}$ in eq. (2.15).) For the SEnKF, we see that this corresponds to tapering the Kalman gain and both approaches boil down to equivalent implementations. The weighting reads

$$\boldsymbol{x}_e^{n,a,b} = (1 - \boldsymbol{w}_b)\boldsymbol{x}_e^{n,a,b-1} + \boldsymbol{w}_b \sum_{j \in \mathcal{B}_b} \boldsymbol{x}_e^{n,a}(j), \tag{2.18}$$

where $\boldsymbol{x}^{n,a,0} = \boldsymbol{x}^{n,f}$ and after the last batch, we set $\boldsymbol{x}^{n,a} = \boldsymbol{x}^{n,a,B}$ to obtain a final analysis state. For the global analysis state, this means that most analysis information is used near observation sites, whereas the forecast with its full spread is retained far away from data. The formulation of covariance location in the form of eq. (2.18) is furthermore convenient since the ETKF analysis scheme can be used as black-box without interference of the localisation weights. From a principled statistical perspective, one could process each individual observation in a serial manner, but the collection in prescribed batches reduces iterations. In doing so, one assimilates the spatial data recursively, similar to the assimilation over time, albeit without the dynamical state evolution because all updates happen at the time when the data gets available. For computational efficiency one again imposes some kind of local routine, and in practice this may rely on GC tapering of the matrices involved. This is commonly done in implementations of Kriging, or in applications with sequential uncertainty reduction, where the analysis can depend on the choice of conditioning order, see, e.g. Nussbaumer et al. (2018).

Complementary to inflation as mentioned in Section 2.2, Zhang et al. (2004) present relaxation-to-prior to counteract overfitting. Introducing a scaling parameter $\phi \in [0, 1]$, the weighting vector can be constructed with neatly integrated relaxation as

$$\boldsymbol{w}_b^{\text{relax}} = \sum_{j \in \mathcal{B}_b} \phi \, \boldsymbol{w}_{\text{loc},j}. \tag{2.19}$$

Here, $\phi = 0$ represents a pure Monte-Carlo simulation while $\phi = 1$ is the previously presented scheme without relaxation.

Algorithm 1 summarizes the ETKF with covariance localisation as we will use it later on. We will refer to it as "SparseObsETKF" in order to avoid mixing up with the LETKF. This implementation help us to keep good control of correlations as well as computational overhead in the presence of sparse observations, and it provides a convenient integration of tapering and relaxation.

Algorithm 1 Analysis scheme with localisation for sparse observations (SparseObsETKF)

Given $\boldsymbol{X}^{n,f}$. Parameters: localisation radius and relaxation ϕ
Set $\boldsymbol{X}^{n,a,0} = \boldsymbol{X}^{n,f}$
for $b = 1, \ldots, B$ **do**
  Allocate $\boldsymbol{w}_b^{\text{relax}}$ ▷ N_X
  for $j \in \mathcal{B}_b$ **do**
    Calculate local $\boldsymbol{X}^{n,a}(j)$ using the ETKF where $\boldsymbol{X}^{n,a,b-1}$ is the forecast ▷ N_X^{loc}
    $\boldsymbol{w}_b^{\text{relax}}$ += $\phi \boldsymbol{w}_{\text{loc},j}$
  end for
  $\boldsymbol{X}^{n,a,b} = (1 - \boldsymbol{w}_b^{\text{relax}})\boldsymbol{X}^{n,a,b-1} + \boldsymbol{w}_b^{\text{relax}} \sum_{j \in \mathcal{B}_b} \boldsymbol{X}^{n,a}(j)$ ▷ N_X
end for
$\boldsymbol{X}^{n,a} = \boldsymbol{X}^{n,a,B}$

3 Comparison against the Analytical Kalman Filter in a Linear Gaussian Advection Diffusion Model

In this section, we examine a linear Gaussian spatio-temporal model. As mentioned in Section 2, this means that the analytic KF in eqs. (2.5) and (2.6) defines the optimal solution. Ensemble-based approximations

and localisation effects of the different filtering techniques from Section 2 can be verified against the KF.

3.1 Advection diffusion model

Inspired by Sigrist et al. (2015), we consider a stochastic advection diffusion equation for state c given by

$$\frac{\partial c(t,\boldsymbol{s})}{\partial t} = \nabla \cdot d\nabla c(t,\boldsymbol{s}) - \boldsymbol{v}_t \cdot \nabla c(t,\boldsymbol{s}) + \zeta c(t,\boldsymbol{s}) + W(t,\boldsymbol{s}). \tag{3.1}$$

The model's parameters are $d = 0.25$ for the diffusion, $\boldsymbol{v} = (1.0, 0.1)^\top$ for the advection, and $\zeta = -0.0001$ for the damping. We assume the stochastic error process W has uncorrelated elements over time but smooth dependent spatial components at each time, and that eq. (3.1) holds for one sampled path (realisation) of W. We consider a rectangular spatial domain $[0, 5] \times [0, 3]$ with periodic boundary conditions, and c will be initialised at time $t = 0$ as a Gaussian random field.

Equation (3.1) can be used to represent for instance marine pollution dynamics (Foss et al., 2021), where the goal is to predict the concentration $c = c(t, \boldsymbol{s})$ of a contaminant over time and space in the ocean. In that case, the advection parameter $\boldsymbol{v}$ would typically come from a full ocean model if vertical currents are ignored.

In the discretised setting, the spatial domain is covered by a uniform Cartesian grid with center points $(\boldsymbol{s}_i)_{i=1}^{N_s}$ in quadratic cells of size 0.1×0.1. The state vector $\boldsymbol{x}^n$ collects all concentrations $c(t^n, \boldsymbol{s}_i)$ at regular time steps t^n. The initial state is represented by $\boldsymbol{x}^0 \sim \mathcal{N}(\boldsymbol{\mu}^0, \boldsymbol{\Sigma}^0)$ with mean vector $\boldsymbol{\mu}^0$ and covariance matrix $\boldsymbol{\Sigma}^0$ having Matérn-type

$$\boldsymbol{\Sigma}^0_{k,l} = \sigma^2(1 + \psi D_{k,l}) \exp(-\psi D_{k,l}),$$

where $\sigma = 0.5$ is the standard deviation (assumed constant at all locations) and $\psi = 3.5$ is the Matérn correlation decay parameter, and $D_{k,l}$ is the distance between $\boldsymbol{s}_k$ and $\boldsymbol{s}_l$. The mean $\boldsymbol{\mu}^0$ equals 10 in the north-east with higher bell-shaped concentration values in the south-west, see Figure 3.2 (left).

For the numerical solution of the SPDE in eq. (3.1), a temporal forward and spatial central finite-difference scheme is employed such that the model resembles eq. (2.1) with the linear operator $\mathcal{M} = \boldsymbol{M}$. With periodic boundary conditions the low-concentration area leaves the domain on the east boundary and enters from the west. The model error $\boldsymbol{\nu}$ is again represented by a Gaussian random field with a covariance matrix $\boldsymbol{Q}$ of a similar Matérn-type. A smaller standard deviation $\sigma = 0.125$ is used, and larger correlation decay parameter $\psi = 7.0$ leading to model noise with smaller correlation.

3.2 Experiment design and analytic solution

A single realisation of the advection diffusion generated by the forward model is used to retrieve observations for the filtering, see Figure 3.1. It is simulated for 250 time steps with $\Delta t = 0.01$ on a grid of size 50×30. The simulated process is observed at $t^n = 25n, n = 1, \ldots, 10$ at 15 grid cells marked red in Figure 3.1. These direct state observations are made with a small observation error $\boldsymbol{\epsilon}^n \sim \mathcal{N}(0, r^2\boldsymbol{I})$, $r = 0.1$.

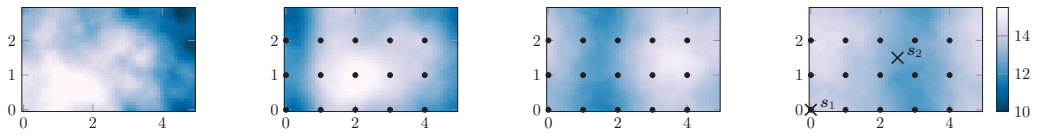


Figure 3.1: The "truth" realisation of the stochastic process at the initialisation and selected observation times with black dots marking the observation sites and black crosses signifying two selected locations of interest.

The KF reference solution is depicted for a selection of time steps in Figure 3.2. As for the truth in Figure 3.1, the filtering mean (Figure 3.2, top) shows an east north-east movement of the concentrations

as expected from the advection term. The standard deviations (Figure 3.2, bottom) are clearly reduced by the data assimilation, especially around the observation sites and in the advection direction. With time, however, the accuracy of the solution converges as the corrections from doing data assimilation are balanced out by the dynamic model errors.

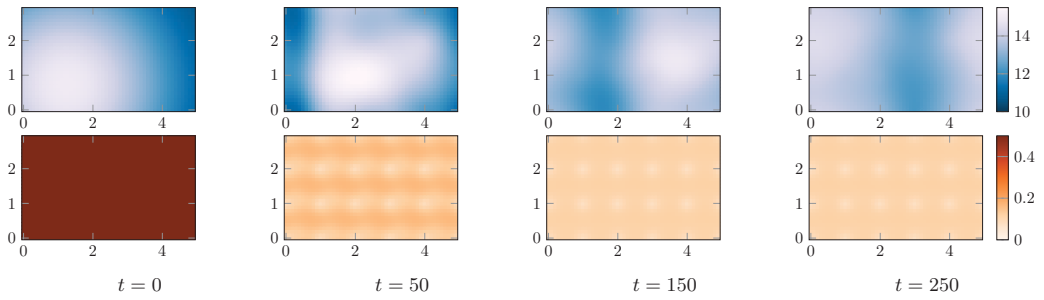


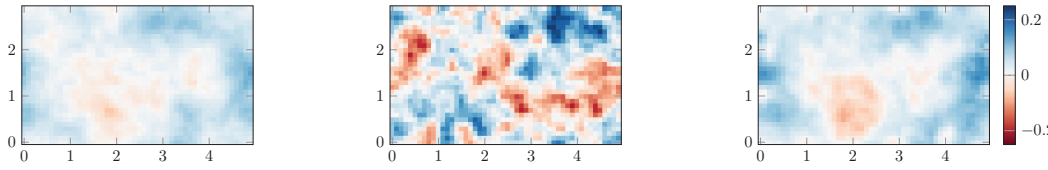
Figure 3.2: For the same times as in Figure 3.1, the resulting analysis mean (upper row) and standard deviation (lower row) of the KF.

3.3 Numerical results and evaluation metrics

We now solve this concentration advection diffusion problem using the ensemble-based methods IEWPF, ETKF, and SparseObsETKF from Section 2. The parameter β emerging in the IEWPF is tuned manually and set to 0.55, and this will be discussed further in relation to some of the results. We set the localisation radius of the SparseObsETKF equal to the correlation range of the model error, which leads to four observational batches. First, we do not use any relaxation in the perfect linear model, as suggested by Raanes et al. (2019). The performance of ensemble-based solutions are opposed to the KF reference solution. We use a set of metrics to evaluate different statistical aspects of the data assimilation methods.

Root-mean-squared error The ensemble mean $\overline{\boldsymbol{x}}^a$ is compared with the KF mean $\boldsymbol{\mu}^a$. Here, we consider the state at $t = 250$ after assimilating all available observations. The error in the mean at each position is then the vector $\mathbf{err}^{\mathrm{KF}}_{\mathrm{mean}} = (\boldsymbol{\mu}^a - \overline{\boldsymbol{x}}^a)$. As a scalar metric to compute the behaviour over all grid cells, we use the root-mean-squared error (RMSE)

$$\mathrm{RMSE} = \|\mathbf{err}^{\mathrm{KF}}_{\mathrm{mean}}\|_2. \tag{3.2}$$



(a) IEWPF [1.76] (b) ETKF [3.35] (c) SparseObsETKF [2.26]

Figure 3.3: Mean error $\mathbf{err}^{\mathrm{KF}}_{\mathrm{mean}}$ at $t = 250$ for assimilation experiments with $N_e = 50$ ensemble members. RMSE is given in the bracket for each specified method.

Figure 3.3 shows $\mathbf{err}_{\mathrm{mean}}$ at each grid cell for a single data assimilation experiment with $N_e = 50$ ensemble members for each method. The RMSEs are the caption brackets. All three ensemble-based data assimilation

methods lead to means that closely resemble the KF reference solution. The mean error is in general low and smoothly distributed for both IEWPF and SparseObsETKF, whereas the errors of EKTF are somewhat bigger. Based on RMSE, IEWPF performs slightly better than SparseObsETKF, whereas the RMSE for ETKF is about twice that of IEWPF.

To deduce reliable conclusions beyond one data set and single ensembles, we repeat the data assimilation experiment multiple times for several independently generated true states. In Table 2, we report averaged results for 20 replicate synthetic truths and five ensemble-based data assimilation experiments each. For this relatively small ensemble size of $N_e = 50$, the localisation in the SparseObsETKF halves the RMSE compared to the standard ETKF, and the RMSE of the IEWPF lies in the middle of the ETKF with and without localisation.

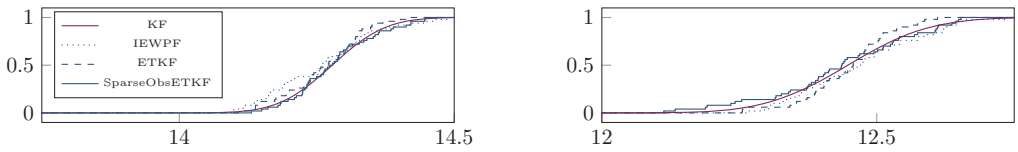
Frobenius covariance difference We contrast the empirical covariance estimates $\widehat{\boldsymbol{\Sigma}}^a$ in eq. (2.11) with the KF reference $\boldsymbol{\Sigma}^a$. We compute the Frobenius covariance difference (FCD) to compare these covariance matrices:

$$\text{FCD} = \|\boldsymbol{\Sigma}^a - \widehat{\boldsymbol{\Sigma}}^a\|_{\text{F}},$$

where $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm (elementwise sum). Averaged results for the FCD over replicate experiments are presented in Table 2 using $N_e = 50$. Here, we see that the FCD for IEWPF and SparseObsETKF are very similar for all cases. The covariance approximations show smaller errors for the ETKF solution than for the other methods. At a single step, the ETKF approximation to the covariance is unbiased, while the other methods have no such guarantee. Still, it is not obvious that the ETKF performs better after many data assimilation steps. Also, when we test the entries close and far from the diagonal of the covariance matrix, we cannot see any other behaviour in the results.

Integrated quadratic distance We next study a metric for the marginal distribution mismatch of discrete ensemble-based distribution approximations to the Gaussian KF reference solution. The reference cumulative distribution function (CDF) of the KF is denoted F^a. The empirical cumulative distribution function (ECDF) of the ensemble-based solutions are denoted $\widehat{F}^a$.

In the analysis two specific locations shown in the far right panel of Figure 3.1 are studied based on their different characteristics: $\boldsymbol{s}_1 = (0, 0)$ is an observation site and $\boldsymbol{s}_2 = (2.5, 1.5)$ is as far away from observation data as possible.



(a) CDF and ECDFs at $\boldsymbol{s}_1$ (b) CDF and ECDFs at $\boldsymbol{s}_2$

Figure 3.4: For two distinct positions, $\boldsymbol{s}_1$ as observation site and $\boldsymbol{s}_2$ far away from observation sites, at $t = 250$ the CDF of the Kalman filter is compared to the ECDFs of IEWPF [0.0242, 0.0238], ETKF [0.0164, 0.0254], and SparseObsETKF [0.0093, 0.0117], respectively, with $N_e = 50$. In brackets, the integrated quadratic difference d_{IQ} for $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$, respectively.

In Figure 3.4, the CDF of the KF is depicted in comparison with the ECDFs of IEWPF, SparseObsETKF, and ETKF at the two different locations for a small ensemble size. First, since the scales of the x-axis in both displays are the same, it becomes obvious that the standard deviation at an observation site is much smaller than at an unobserved location. Next, we see that the different filtering methods differ in quality when compared to the analytic solution. For the observation site $\boldsymbol{s}_1$, there is no clear qualitative difference, but at $\boldsymbol{s}_2$ one may already identify a slight divergence in ETKF's ECDF, whereas SparseObsETKF and IEWPF still approximate the reference CDF quite well.

The tuning parameter β steers the spread in the analysis ensemble of the IEWPF and among the presented evaluation metrics the ECDF reveals the scale the best. We used it to optimise the filtering distribution manually and found 0.55 as best choice. For smaller values, the ECDF gets too sharp and for higher values the spread gets too large. Similarly, the variance in the SparseObsETKF-ensemble usually increases as the relaxation parameter ϕ decreases. When using $\phi < 1$ we observed that the spread in the ensemble becomes too big compared to the CDF and the best match is achieved for $\phi = 1$.

Thorarinsdottir et al. (2013) suggest a proper divergence function to compare marginal CDFs, condensing the error into a scalar number. The integrated quadratic difference is defined by

$$d_{\mathrm{IQ}} = \int \left(F^a - \widehat{F}^a \right)^2 \mathrm{d}x,$$

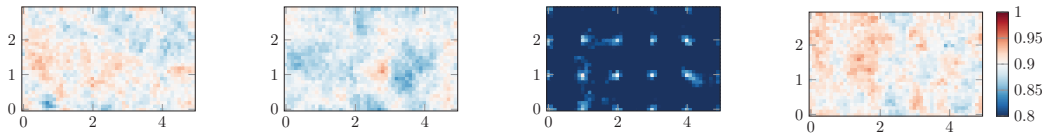
where the quadratic error is integrated over the sample space of the variable. Errors captured in d_{IQ} can originate from either a lack of Gaussianity, or a wrong scaling or a bias, or a combination.

Table 2 shows averaged results for d_{IQ} at $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ for the three ensemble methods. IEWPF and ETKF produce similar results, while SparseObsETKF clearly gives the best results. The reason is twofold: First, the IEWPF and ETKF update the entire field at each data assimilation time, even at locations like $\boldsymbol{s}_2$ that are far away. With the limited ensemble size, this is likely to induce some undesired bias and variability far from data. Second, the SparseObsETKF is rather accurate near the observations sites, like the other filters, and because the advection and diffusion are known, the local updating propagates reasonably over time to the far location $\boldsymbol{s}_2$.

Probability coverage level Based on the mean and variance of the ensemble-based solutions, we can check how often a prediction interval covers the true realisation. For the KF reference, we will have near nominal coverage because the truth is simulated from the same model. Coverage probabilities (CPs) of the analysis after the first observation time are

$$CP^1_{1.64} \coloneqq P\Big(\boldsymbol{x}^1_{\mathrm{true}} \in \left[\boldsymbol{\mu}^{1,a} \pm 1.64\boldsymbol{\sigma}^{1,a}\right]\Big) \approx 0.90.$$

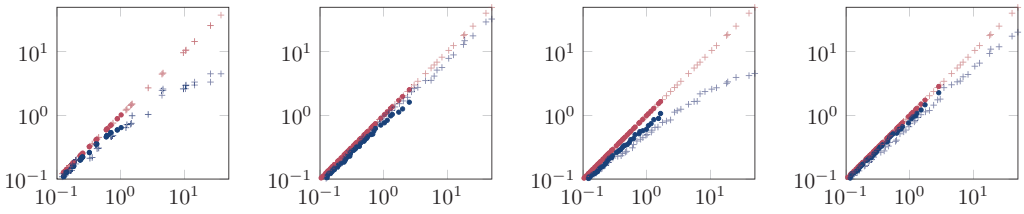
This means that the probability that the truth is covered in the interval of 1.64 standard deviations from the mean is 90%. For all methods, we use replicated synthetic truths, and estimate the CPs.



(a) KF [89.7%] (b) IEWPF [88.7%] (c) ETKF [77.8%] (d) SparseObsETKF [90.4%]

Figure 3.5: Estimated $CP^1_{1.64}$ using 500 replication experiments for the KF and the ensemble-based methods with $N_e = 50$. The brackets show the averaged estimated coverage probabilities. The color scale is centered at the target probability of 90%.

Figure 3.5 shows the estimates of the CPs averaged over 500 runs. The KF results cover the nominal 90% very well, with a variability as expected from 500 replicates. Even though that the KF is analytic, the truths are generated stochastically and we see the Monte Carlo error here. The IEWPF and the SparseObsETKF also give very good estimates even though the ensemble size of 50 members is low, and there are no visible structures around observation sites. In contrast, the ETKF without localisation suffers from strong under-coverage in this experiment. The CPs are around 0.9 near observation sites but fall to lower levels, which are outside the truncation interval of the plot, away from these.



(a) KF (b) IEWPF (c) ETKF (d) SparseObsETKF

Figure 3.6: Eigenvalue update in the very first (crosses) and very last (dots) data assimilation step. Prior-vs-prior in red, prior-vs-posterior in blue.

Eigenvalues analysis An eigendecomposition of the covariance matrix yields eigenvalues representing the variability among orthogonal axes of linear combinations of state variables. The largest (first) eigenvalue is the variance in the direction of the first eigenvector, and for a Gaussian distribution this represents the largest half-axis in the ellipsoid defined via the quadratic form. Further, for a Gaussian distribution, the entropy (disorder) is defined via the log-determinant of the covariance matrix which is the sum of the log-eigenvalues. By studying the eigenvalues of the covariance matrices of the different data assimilation methods, we hence gain insight in the variability reduction and the disorder of the distributions. Since it does not make sense to average eigenvalues over multiple realisation, we leave the spectral analysis to a qualitative view with cross-plots of eigenvalues for the different methods.

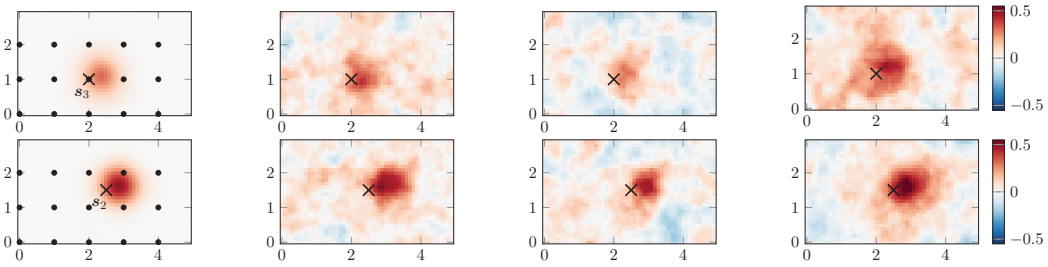
Figure 3.6 shows cross-plots of the covariance matrix eigenvalues of forecast (first axis) and assimilated (second axis). In all displays, the crosses represent the first data assimilation step while the dots are at the last data assimilation step. For the KF, eigenvalues are computed directly from the covariance matrix and for the ETKF variants this is the eigenvalues of the estimated covariance matrix in eq. (2.11). In the case of the IEWPF, the prior ensemble is without model error and hence we perturb the prior before plotting.

In Figure 3.6, the forecast-vs-forecast points (red) lie on a straight line, but we still notice that all the red dots are closer to the origin than some of the crosses. Hence, the eigenvalues are clearly smaller at the last step, indicating that the data assimilation over time gives smaller entropy. Going from forecast (red) to assimilated (blue), the eigenvalues of the covariance matrix are reduced. Taking the KF as a benchmark, this reduction is particularly large for the biggest eigenvalues, indicating the updating is not only local but also shrinks the variability of dominating linear combinations and the entropy of the distribution. At the first step (crosses), the ETKF updating appears very similar to that of the KF. At the last step, the eigenvalues of the ETKF are larger and not reduced quite like for the KF. This indicates that even though the ETKF undercovers the distribution (Figure 3.5), there is not quite sufficient reduction in the largest eigenvalues. Both IEWPF and SparseObsETKF get smaller reduction in eigenvalues than the KF. At the first data assimilation step the reduction is larger for SparseObsETKF than for the IEWPF, while the IEWPF has more reduction at the last data assimilation step.

Spatial connectivity While the previous metrics have considered the marginal solution at one time step only, the correlation between different time steps and between different spatial positions gives further insight into the statistical quality of the filtering methods.

Given data up to time t^{n-1}, the correlation between the concentration at $\boldsymbol{s}_k$ at $t^{n-1} = 225$ and $\boldsymbol{s}_l$ at t^n can then be calculated from the KF results via

$$\mathbb{C}\text{orr}\left(\boldsymbol{x}_k^{n-1,a}, \boldsymbol{x}_l^{n,f}\right) = \frac{\mathbb{C}\text{ov}\left(\boldsymbol{x}_k^{n-1,a}, \boldsymbol{x}_l^{n,f}\right)}{\sigma_k^{a,n-1}\sigma_l^{f,n}} = \frac{\boldsymbol{M}\boldsymbol{\Sigma}_{k,l}^{a,n-1}}{\sqrt{\boldsymbol{\Sigma}_{k,k}^{a,n-1}}\sqrt{\boldsymbol{\Sigma}_{l,l}^{f,n}}}. \tag{3.3}$$



(a) KF (b) IEWPF [2.73, 2.71] (c) ETKF [2.32, 2.64] (d) SparseObsETKF [3.86, 3.13]

Figure 3.7: Correlations $\mathbb{C}\mathrm{orr}\left(\boldsymbol{x}_k^9, \boldsymbol{x}_l^{10}\right)$ and $\widehat{\mathbb{C}\mathrm{orr}}\left(\boldsymbol{x}_k^9, \boldsymbol{x}_l^{10}\right)$ between fixed locations $\boldsymbol{s}_k$ and all other grid points in the domain $\boldsymbol{s}_l$, for $k = 3$ in the upper and $k = 2$ in the lower row, respectively. The ensemble-based methods use $N_e = 250$ ensemble members, and the respective CE estimates are given in the square brackets.

Similarly, we can estimate these correlations from the ensemble-based methods by

$$\widehat{\mathbb{C}\mathrm{orr}}\left(\boldsymbol{x}_k^{n-1,a}, \boldsymbol{x}_l^{n,f}\right) = \frac{1}{N_e - 1} \frac{1}{\hat{\sigma}_k^{n-1,a} \hat{\sigma}_l^{n,f}} \sum_{e=1}^{N_e} (\boldsymbol{x}_{e,k}^{n-1,a} - \overline{\boldsymbol{x}}_k^{n-1,a})(\boldsymbol{x}_{e,l}^{n,f} - \overline{\boldsymbol{x}}_l^{n,f}).$$

In Figure 3.7, we show the correlation fields of the state between a reference grid cell at $t^9 = 225$ and all other grid cells at $t^{10} = 250$. As reference locations $\boldsymbol{s}_k$, we consider $\boldsymbol{s}_2$ positioned far away from any observations in the top row, and $\boldsymbol{s}_3$ at an observation site in the middle of the domain displayed in the lower row. First, from the KF solution (left), we recognise the advection field in the model that transports information towards east-north-east from both locations, as well as the diffusion causing the correlation to have longer range than the model error correlation radius. Second, the maximal correlation to the reference point is higher when $\boldsymbol{s}_k$ is not an observation site. In spatial statistics, conditioning on data breaks up some of the prior correlations. Since most of the update from the data assimilation occurs near the observation locations, the conditional correlation tends to be smaller in the proximity of data. At locations that are far from observations, more of the prior correlation remains.

In the three rightmost columns of Figure 3.7, we see the correlations estimated with the three ensemble-based methods using an ensemble with $N_e = 250$ members. We see that all methods capture a similar correlation structure with respect to the advection and diffusion, and the relative balance between prior model and information from the observation. With $N_e = 250$, results are less smooth than the KF solution. In the upper scenario, the area of high analytical correlations becomes less apparent among background noise, while in the lower scenario the respective regions are easier to identify in all methods.

The spatial error in the approximation of correlation between two consecutive model steps for the reference location $\boldsymbol{s}_k$ is evaluated collectively across all grid cells as

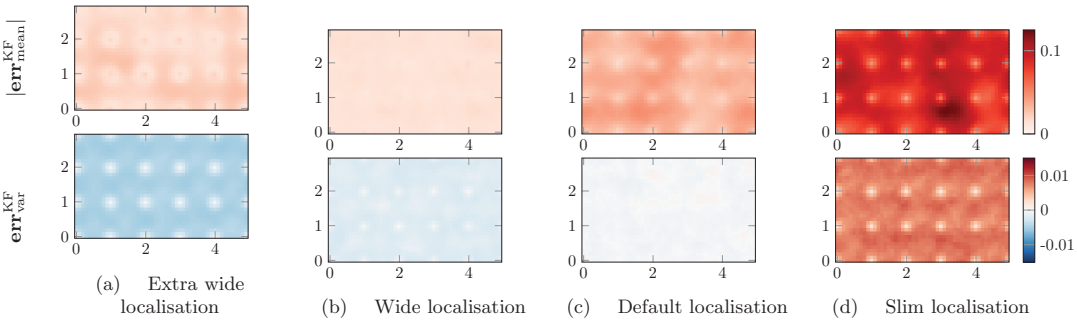
$$\mathrm{CE}(\boldsymbol{s}_k)^2 = \sum_{l=1}^{N_s} \left| \mathbb{C}\mathrm{orr}\left(\boldsymbol{x}_k^{n-1,a}, \boldsymbol{x}_l^{n,f}\right) - \widehat{\mathbb{C}\mathrm{orr}}\left(\boldsymbol{x}_k^{n-1,a}, \boldsymbol{x}_l^{n,f}\right) \right|^2. \tag{3.4}$$

The CE for the specific data assimilation run shown in Figure 3.7 are given in square brackets in the figure captions. These results for a single run with $N_e = 250$ already reveal that the contribution to CE can come from multiple sources, such as over- or under-estimation of the actual correlations and from spurious correlations. The final CE does not qualitatively expose which of the error sources are present to which extent, but it quantifies how well the analytical structure is approximated. The chosen $\mathbb{C}\mathrm{orr}$ and $\widehat{\mathbb{C}\mathrm{orr}}$ suggest that spurious correlations are a present error source in all methods, but IEWPF and SparseObsETKF tend to

overestimate the high correlations. Meanwhile, ETKF underestimates the correlations, thus leading to a smaller CE.

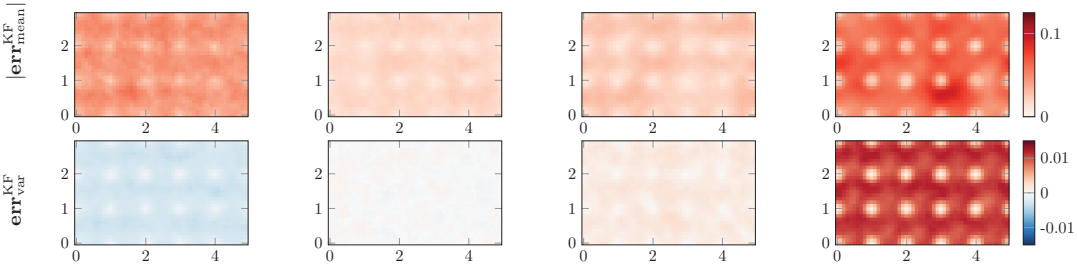
3.4 Sensitivity to localisation

In the IEWPF the localisation properties are steered by the structure of the model error covariance $\boldsymbol{Q}$ and in the SparseObsETKF by the definition of the localisation radius, which we again defined according to the correlation range of $\boldsymbol{Q}$, see Section 2.3. In the rest of the section those choices are hold fixed, but here we showcase their influence. To do so, we consider four different ψ for the model error covariance matrix, while keeping everything else unaltered, i.e. the standard derivation and the IEWPF tuning parameter β. Analogously, we vary the localisation radius in the SparseObsETKF. The parameter $\psi = 3.0$ corresponds to no localisation and $\psi = 5.0, 11.0$ represent roughly a doubling and halving of the localisation radius, respectively. Importantly, it should be noted that the GC and the Matérn-type covariance kernel decay with different rates such that the results are not one-to-one comparable, but we can still record trends within each method.



(a) Extra wide localisation (b) Wide localisation (c) Default localisation (d) Slim localisation

Figure 3.8: Different localisation parametrisations for the IEWPF. Averaged results over 20 truth realisations and 5 ensemble initialisation each.



(a) No localisation (b) Wide localisation (c) Default localisation (d) Slim localisation

Figure 3.9: Different localisation parametrisations for the SparseObsETKF. Averaged results over 20 truth realisations and 5 ensemble initialisation each.

Figure 3.8 and Figure 3.9 allow to get an impression of the RMSE together with spatial effects. Since the localisation is in-build into the IWEPF, one has to be careful with the interpretation of the left-most

Table 2: Metrics for marginal distribution averaged over 20 synthetic truths and 5 ensemble realisations, each with $N_e = 50$. Standard deviations are given in brackets.

	RMSE	FCD	d_{IQ} at $\boldsymbol{s}_1$	d_{IQ} at $\boldsymbol{s}_2$
MC	8.27 [2.88]	47.0 [8.35]	12.8E-02	13.5E-02
IEWPF	1.67 [0.43]	2.77 [0.14]	2.51E-02	2.58E-02
ETKF	2.14 [0.40]	2.14 [0.04]	2.57E-02	2.86E-02
SparseObsETKF	1.15 [0.24]	2.79 [0.15]	1.29E-02	1.68E-02

row (a). Nevertheless, there are clear unintended artefacts in mean and variance errors. Even though the mean for $\psi = 5.0$ in row (b) is very well calibrated in the entire field, the variance is slightly underestimated and the error shows structures around observation locations. Note that an increase of β may counteract the underestimation. For the standard IEWPF, this is the opposite way around, here the error in the variances is minuscules, but the mean is not eqully well calibrated. The IEWPF with reduced localisation radius performs poorly when further away from observation locations. The ETKF without localisation underestimates the variance of the KF. The SparseObsETKF with a localisation radius that spans the full y-extent of the domain (row (b)) performs for mean and variance estimation slightly better than the parametrisation as we have chosen in the rest of the section. We remind that the usually used localisation yields a computational advantage due to reduced analysis dimensions. Again the smallest localisation parametrisation fails to assimilate the entire domain.

We note that the localisation parametrisation as used in the rest of the section work reasonably well. Nevertheless we notice that there is potential to fine-tune both methods further, but in the interest of a limited number of tuning parameters we continue with the native parameters.

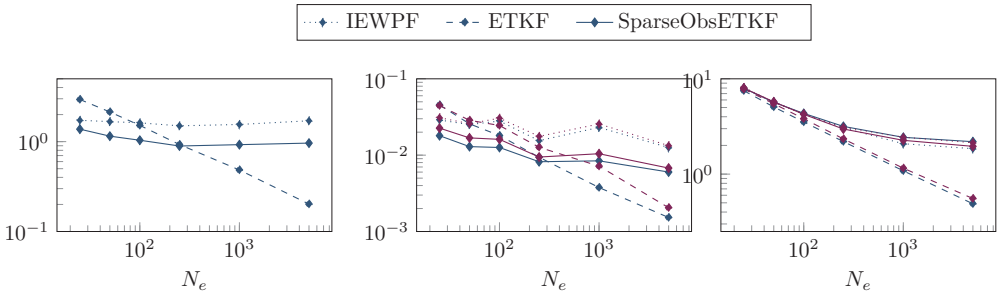
3.5 Discussion of evaluation metrics

The set of comparative metrics from the previous subsection has given us a collection of metrics that quantify some statistical qualities of the ensemble-based data assimilation methods in reference to the analytical KF solution.

Table 2 shows the statistically averaged results for these performance scores at $t = 250$. These results are obtained across five data assimilation runs for 20 different synthetic truths and are therefore more reliable than the single realisations demonstrated in Figures 3.3, 3.4 and 3.7. We have here used $N_e = 50$ ensemble members for each run. In addition to comparing the data assimilation methods against each other, we have also included the results using pure Monte Carlo simulations without observations (top row). These serve to demonstrate the worst-case scenario for each metric, and we see how all three data assimilation methods clearly outperform this, as expected. In the experiments, we have observed that the IEWPF takes several assimilation steps until it is sufficiently calibrated, what is respected by the the choice of t here such that the comparison stays fair, see Section 4 for details.

Based on the results in Table 2 there is no method that clearly dominate on all individual criteria. For RMSE, it seems that SparseObsETKF is much better than ETKF, but this is not as clear when considering FCD, where ETKF scores best. Maybe more surprising, SparseObsETKF is significantly better than ETKF when measuring the error in the ECDF at $\boldsymbol{s}_2$ far from the observation, but not at $\boldsymbol{s}_1$ at an observation site. A plausible explanation is that when updating the state far from an observation, all covariances are relatively weak, which means that spurious correlations more easily dominate data assimilation. With localisation, we ensure that only the most relevant small correlations are considered, thus improving the result. This effect will then be less at an observation site as the most important correlations are stronger. IEWPF gets all metrics between ETKF and SparseObsETKF. We observe that worse FCD has no influence on the d_{IQ} at the considered positions.

Effects of ensemble size In Figure 3.10, we study how different ensemble sizes influence RMSE, d_{IQ}, and correlations for the three ensemble-based methods. We use ensemble sizes $N_e \in \{25, 50, 100, 250, 1000, 5000\}$,



(a) RMSE

(b) d_{IQ} at $\boldsymbol{s}_0$ (blue) and $\boldsymbol{s}_2$ (purple)

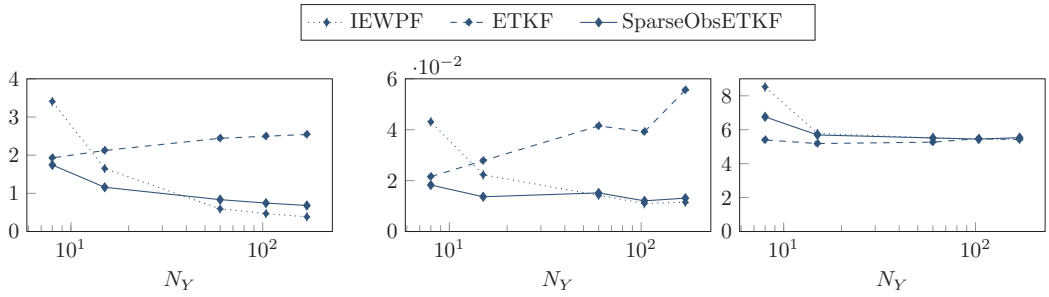
(c) $\mathrm{CE}(\boldsymbol{s}_3)$ (blue) and $\mathrm{CE}(\boldsymbol{s}_2)$ (purple)

Figure 3.10: Evolution of the comparison measures in depends of the ensemble size N_e.

and the results are averaged across multiple experiments for each of these sizes. In general, we expect that increased ensemble size leads to more accurate statistical estimates and thereby better results. This is clearly the case for ETKF for all metrics, and SparseObsETKF shows the same trend, but not as strongly. We see that SparseObsETKF outperforms ETKF with respect to RMSE and d_{IQ} for small ensemble sizes, but ETKF is better with large ensembles as the performance of SparseObsETKF stagnates for $N_e > 250$. SparseObsETKF improves less than ETKF with larger ensembles because it ignores correlations, and this gives bias in the analysis. The IEWPF yields results between ETKF and SparseObsETKF for small sample sizes, but there is slower convergence as the ensemble size increases. Unlike ETKF, which converges to the true Gaussian distribution in this case, there is no such guarantee for the IEWPF. Since the second stage perturbation step of the IEWPF is designed to reduce a systematic bias and help performance, fine-tuning the choice of β scaling parameter could improve convergence for some properties, but maybe not similarly so for all the desired scores. The correlations mismatch compared with the KF in Figure 3.7 are slightly different depending on the fixed reference point, but they converge with increasing sample, especially so for the ETKF which has curves going faster to 0. For both SparseObsETKF and IEWPF there seem to be a remaining mismatch in this CE score even for thousands of ensemble members.

Effects of sparsity of observational data In a regime dominated by the sparsity of observations, we also want to stress-test all methods with respect to the amount of observational data. For this purpose, we repeat the case study using $N_Y \in \{8, 15, 60, 104, 170\}$ regularly placed observation sites. These numbers are chosen such that the observation locations have distance of 15, 10, 5, 4, and 3 grid cells apart from each others, respectively. We use $N_e = 50$ ensemble members. Of course, the localisation scheme for the SparseObsETKF is not designed for dense data and will get computationally very inefficient due to a high number of batches that are processed serially. The localisation radius is not modified.

Figure 3.11 shows the same averaged metrics as before, with respect to a growing number of observations. Note that it no longer makes sense to distinguish between locations near and far from observations, since the observation sites get denser over the domain. The increase in observation data leads to a sharpening in the reference distribution calculated from the KF. For the ETKF, we observe that RMSE does not improve and its ECDF approximation gets worse, compared with the KF. This is because of the underestimation in variance and a slight bias which strongly penalises the d_{IQ}. Both SparseObsETKF and IEWPF improve their quality for increasing observation data size. This is surprising and noteworthy for IEWPF, as PFs tends to collapse for high dimensional observations. In contrast to RMSE and d_{IQ}, the CE does not depend on the observation sparsity and is practically constant on the level that we saw in Figure 3.10c.



(a) RMSE (b) d_{IQ} at $\boldsymbol{s}_0$ (c) CE($\boldsymbol{s}_3$)

Figure 3.11: Evolution of the comparison measures as we change the sparsity/density of observation data through the number observations N_Y. The ensemble size is fixed at $N_e = 50$.

Summary In this case study, we verify the ensemble-based methods from Section 2 for a linear Gaussian data assimilation problem with the analytical KF reference. The SparseObsETKF and IEWPF include localisation, either explicitly or implicitly, which is connected to the correlation radius of the model error. Beyond the verification, we can in particular record that SparseObsETKF outperforms ETKF and IEWPF for smaller ensemble sizes (about $N_e \leq 250$). While the ETKF reduces the spectral radius in every data assimilation step more than the other methods, it requires a large ensemble size to obtain a reasonable approximation of the full covariance matrix. The localised version that ignores large-distance correlations is performing well for small ensemble sizes, but it does not improve much more for larger sizes. Similar tendencies are seen with the IEWPF. The approximation of the correlations between different time steps depend mostly on the ensemble size - the model error plays a major role in the evaluation and this criterion requires a higher ensemble size for a sufficient representation. For reasonable ensemble sizes, say 100, both IEWPF and SparseObsETKF operate well for any density of observation data. For most criteria we tested in this example with sparse point data, the SparseObsETKF tends to give slightly better performance than the IEWPF. Based on this extensive statistical evaluation, we hence recommend considering the SparseObsETKF for similar kinds of applications with sparse data and limited ensemble sizes.

4 Comparison for Drift Trajectory Forecasting in a Simplified Ocean Model

We now increase both dimensionality and complexity as we turn to a non-linear simplified ocean model. This gives insight into the behaviour of the ensemble-based data assimilation methods on a challenging case with applied relevance. The practical purpose of this configuration is to use ensembles of computationally efficient simplified ocean models instead of or complementary to single realisations of complex operational ocean models in time critical situations. The simplified models allow for larger ensembles and hence facilitate uncertainty quantification. Such an approach can be useful in search-and-rescue operations, where drifters released by the vessel or relevant anchored buoys (also called moorings) can give sparse in-situ observations during the operation. These point observations can then be assimilated into the ensemble-based representation to improve the drift trajectory forecasts that specify a search area.

Due to the non-linearity of such an model, there is no analytical reference solution for the ensemble distributions available. We can nevertheless compare SparseObsETKF and IEWPF by studying their predictive properties with the ground truth in a simulation study. We base our numerical experiments on those presented in Holm et al. (2020), where the IEWPF was successfully tailored for efficient GPU-accelerated assimilation of point observations of a chaotic shallow-water model. We expand on the numerical result

from that work by evaluating more skill scores, and by providing an in-depth comparison between IEWPF and SparseObsETKF for state estimation and drift trajectory forecasts. In particular, this will also serve as more thorough evaluation of the applicability of IEWPF in this context.

4.1 Simplified ocean model

The simplified ocean model is represented by the rotational shallow-water equations given by

$$\begin{aligned} \eta_t + (hu)_x + (hv)_y &= 0 \\ (hu)_t + \left(hu^2 + \frac{1}{2}gh^2\right)_x + (huv)_y &= fhv \\ (hu)_t + (huv)_x + \left(hv^2 + \frac{1}{2}gh^2\right)_y &= -fhv. \end{aligned} \tag{4.1}$$

This is a non-linear two-dimensional hyperbolic conservation law, which models conservation of mass through the deviation η from equilibrium sea level, and conservation of momentum through hu and hv, which are vertically integrated ocean currents in x- and y-direction, respectively. By denoting the equilibrium depth of the ocean by H, we get the total depth as $h = H + \eta$. Furthermore, g is the gravitational constant and f is the Coriolis parameter that accounts for the rotating frame of reference.

We solve eq. (4.1) using the high-resolution central-upwind finite-volume scheme proposed by Chertock et al. (2018). In our notation from the model equation eq. (2.1), the state vector $\boldsymbol{x}^n$ consists of the cell averaged values $(\eta_i^n, (hu)_i^n, (hv)_i^n)$ at time t^n for all cells i in the discretised domain. The $\mathcal{M}^n$ operator then applies the finite-volume scheme to evolve the state from $\boldsymbol{x}^{n-1}$ to $\boldsymbol{x}^n$. Note that the time step used by the numerical method can be chosen independently from the model time step, meaning that $\mathcal{M}$ might consist of multiple iterations of the numerical scheme.

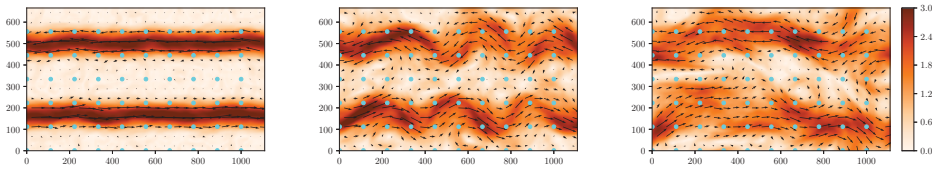
We apply a small-scale Gaussian model error $\boldsymbol{\nu} \sim \mathcal{N}(0, \boldsymbol{Q})$. It is constructed from a coarse-scale perturbation of η, which is smoothed by a second-order autoregressive function and projected onto the numerical grid. The model error for hu and hv is then inferred according to geostrophic balance to ensure physical feasibility. Further details about this model are available in Brodtkorb and Holm (2021) and Holm et al. (2020).

4.2 Experiment design

In the following, we use the same experimental design as in Holm et al. (2020). We consider a rectangular domain covering 1100 km × 666 km that is discretised as a uniform Cartesian 500×300 grid. The domain has periodic boundary conditions and constant equilibrium depth $H = 230.0$ m. The initial conditions, for the ground truth as well as for all ensemble members, consist of a westward jet in the north part of the domain and an eastward jet in the south, with $hv = 0$. Both jets are balanced according to geostrophy by η so that the initial conditions are in steady state. This steady state is however unstable, and slight perturbations, such as those from the model error $\boldsymbol{\nu}$, cause chaotic behaviour.

As an example of the turbulent behaviour, Figure 4.1 shows the water velocities for one realisation that is labeled as the synthetic truth $\boldsymbol{x}_{\text{true}}$. Here, the model error is added every 60 s, and the model error correlation radius is approximately 40 km. From Figure 4.1, we see that the jets in $\boldsymbol{x}_{\text{true}}$ are still quite regular after 3 days, but grow more irregular after 6 and 10 days. It should be noted that the mean state from a pure Monte Carlo experiment without data assimilation will results in $hv \approx 0$ even after 10 days. This indicates that it is challenging to correctly capture where and how the turbulent behaviour will develop.

From $\boldsymbol{x}_{\text{true}}$, we extract direct observations of only (hu, hv) at 60 locations in the domain every 5 minutes between day 3 and day 10, with observational noise sampled from $N(0, \boldsymbol{I})$. The turquoise dots in Figure 4.1 show the observation sites. In total, the experiment is characterised by 450.000 state variables versus only 120 very sparse noisy observations. After day 10, three drifters are released in the domain, and advected according to the simulated currents at every time step of the numerical scheme using a simple Euler scheme. Part of the challenge for the data assimilation methods is to forecast the trajectories of these drifters.



(a) Day 3 (b) Day 6 (c) Day 10

Figure 4.1: State of the synthetic truth after 3, 6, and 10 simulation days. The arrows indicate the direction as well as strength of the ocean currents derived from u and v, respectively. The background visualises the magnitude of velocity in $\frac{\mathrm{m}}{\mathrm{s}}$. The turquoise dots mark the fixed-point buoy positions.

The data assimilation start at simulation day 3 after each ensemble member has been spun up from the steady state through independently sampled model errors. Even though all ensemble members are visually very similar at this stage, they have started to develop internal instabilities that will grow over time unless the observations are successfully assimilated.

This case is much more challenging than the advection diffusion model in Section 3: The shallow-water model is non-linear, there are unobserved variables, and it has significantly higher dimensionality. Critically, the non-linear dynamics of the shallow-water model is challenging to capture. In the advection diffusion model, the state converges towards an equilibrium due to the diffusion, whereas our shallow-water equation case gets chaotic dynamics that makes the ensemble naturally diverge in time.

4.3 Numerical results

Classical EnKF approaches like the ETKF lead to useless results for this difficult case, and only results of the IEWPF and SparseObsETKF are shown in the comparison. We use $N_e = 100$ as a compromise between computational effort and statistical quality. Based on our experiments on this high-dimensional non-linear model, the IEWPF performance is not very sensitive to the explicit choice of β and we use the maximal allowed value. The localisation radius for the SparseObsETKF is chosen slightly larger than the model error correlation radius. In contrast to Section 3, we now also investigate the influence of relaxation in the SparseObsETKF. We present results for the SparseObsETKF without relaxation ($\phi = 1.0$) and for the SparseObsETKF whose weights in the localisation are scaled by $\phi = 0.5$. We compare data assimilation methods with the simulated truth using a number of skill scores that refer to this ground truth.

State estimation We first look at deviations of the ensemble mean from the truth by

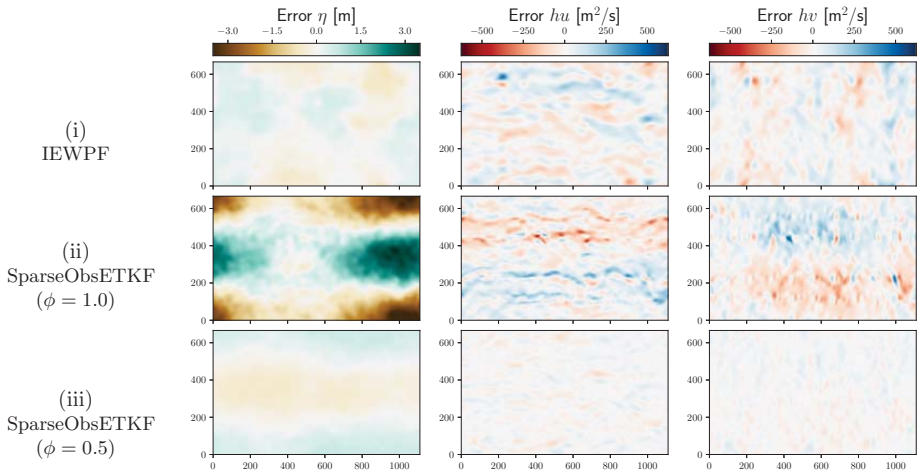
$$\mathbf{err}_{\mathrm{mean}}^{\mathrm{day10,\ true}} = \overline{\boldsymbol{x}}^{\mathrm{day10}} - \boldsymbol{x}_{\mathrm{true}}^{\mathrm{day10}}, \tag{4.2}$$

which represents the error in the correct physical unit. We also investigate the standard deviation in the ensemble

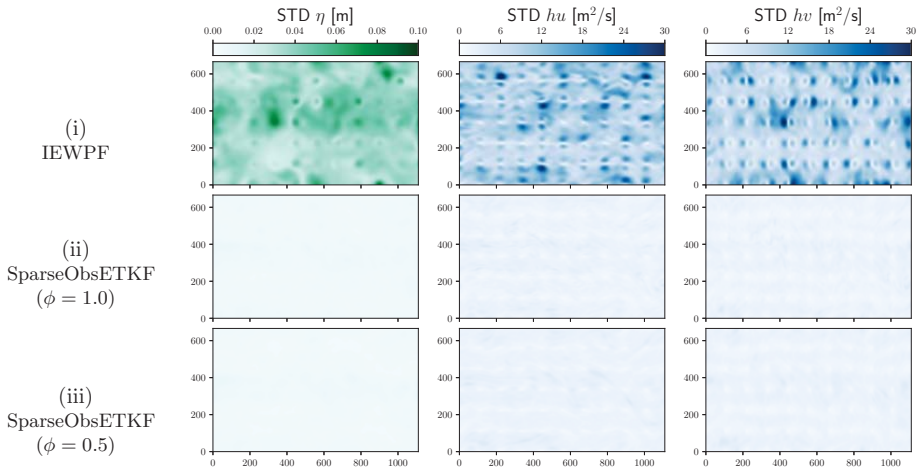
$$\mathrm{STD}^{\mathrm{day10}} = \frac{1}{N_e - 1} \sqrt{\sum_{e=1}^{N_e} \left(\boldsymbol{x}_e^{\mathrm{day10}} - \overline{\boldsymbol{x}}^{\mathrm{day10}} \right)^2}, \tag{4.3}$$

which gives insight about the ensemble spread around its mean.

Figure 4.2a shows the mismatch between the truth and the ensemble means of the conserved variables after assimilating the final observations on day 10. Significant differences become clear in the error of the sea-surface elevation η (left): While the IEWPF has some moderate, relatively smooth error over the entire domain, the mean of the SparseObsETKF is far off in half of the domain. In particular, the rims in the error field are very sharp, also recognisable in the error spots of the currents at the edges of the jets. This



(a) Error $\mathbf{err}_{\text{mean}}^{\text{day10, true}}$



(b) Standard deviation $\text{STD}^{\text{day10}}$

Figure 4.2: Properties of the state estimation for the different physical variables in this simplified ocean model (sea-surface elevation η as well as momenta hu and hv) after day 10 measured in the error of the mean versus the truth and the standard deviation presented for the IEWPF as well as SparseObsETKF without and with inflation.

indicates that the ensemble produces very fast changing ocean fields with the tendency to non-physical members. However, relaxation with $\phi = 0.5$ (bottom) impressively fixes some of those issues and the error fields become much smoother and closely calibrated, even though there is still a recognisable, but weak inherited pattern in the error for elevation.

There are structured artefacts identifiable around the observation sites for IEWPF in Figure 4.2b. Even though Figure 4.2a shows that the ensemble mean gives a very precise description of the ground truth, the ensemble variance is large. As discussed in Holm et al. (2020), the IEWPF updates the momentum locally by inducing a corrective current formed by the structure in the model error covariance matrix $\boldsymbol{Q}$. In this case, $\boldsymbol{Q}$ induces geostrophically balanced dipole structures, which means that while improving the state at the observation site, we risk deteriorating the solution in its vicinity. This illustrates a weakness of IEWPF, showing that its quality is only as good as the structure of $\boldsymbol{Q}$.

In the standard deviations for the SparseObsETKF in Figure 4.2b (middle row), there is an expected pattern of low values around observation sites. Since the localisation only corrects around the buoys and leaves the forecast otherwise unchanged, the variance in one data assimilation step is mainly reduced in local areas. With the dynamical model over time, the variance reduction is disseminated over the entire domain. Furthermore, the standard deviation in the SparseObsETKF is on a very low level. Having areas of low error together with the sudden changes towards big errors suspects overfitting. Also the relaxed SparseObsETKF (lower row) achieves a similarly low standard deviation after day 10.

Drift trajectory forecasting To further compare the practical applicability of IEWPF and SparseObsETKF, we look at forecasts of drift trajectories starting at day 10.

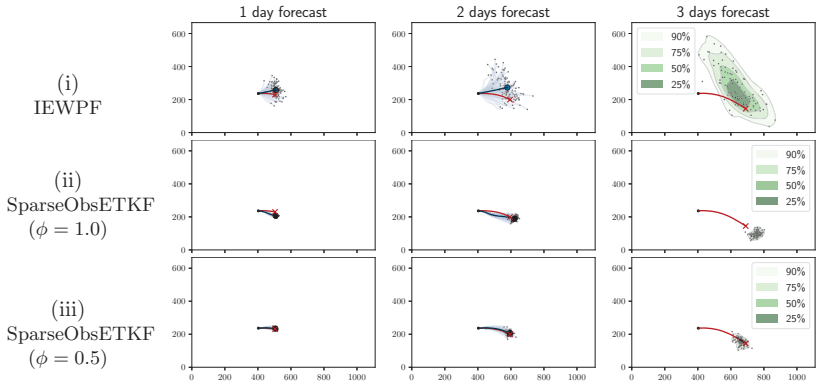
Figure 4.3 demonstrates the forecasted trajectories of drifters that are realised after ten simulation days in the simplified ocean model. The three drop locations are selected to capture different characteristics in the currents: Drifter 1 (display (a)) starts in the middle of a rather weak and big east stream. Drifter 2 (display (b)) starts in a rather strong west stream and drifter 3 (display (c)) starts in a turbulent area in between the dominating streams. For the first two days of forecast, we show the true trajectory along with the trajectories for all ensemble members and the ensemble mean, whereas for the third day (right) we show the estimated kernel density (Scott, 1992) of the final drifter locations along with the true trajectory.

For drifter 1, all trajectories have an east-wards drift, but the IEWPF members fan out from the beginning while the SparseObsETKF trajectories stay close together. Without relaxation, the truth becomes an outlier in the SparseObsETKF forecast. With relaxation, the truth stays within the forecast. The trajectories from the IEWPF catch the truth in a high-probability area, but their spread covers almost the entire extent of the domain in the y-direction.

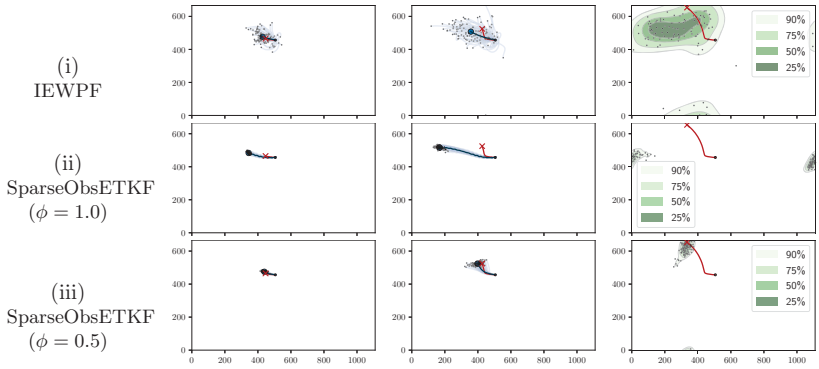
Even though drifter 2 starts within a jet, it drifts only shortly west-wards before it takes a sudden turn towards the north. Here, we can again see the turbulent behaviour of this non-linear model. The trajectories of the IEWPF again spread out widely, and therefore does not reveal any consistent dynamical pattern in the underlying currents. SparseObsETKF not only misses the true trajectory completely, it also shows some wriggling trajectories which indicates that there are unbalanced gravitational waves in the ensemble. Relaxation increases the spread in the trajectories up to day 2, and most of the ensemble members capture the sudden turn in the truth, even though this happens a day after assimilating the final observations.

Drifter 3, which is released in an unstable area, follows what is almost a rotation-like pattern. Here, the IEWPF is unable to estimate a clear direction even for the first 24 hours, and after day 10 the drifter distribution stretches out across almost half the simulation domain. In contrast, SparseObsETKF with and without relaxation the ensemble gives a precise forecast for the first day, only showing a spread for the two last days. The truth is well represented by the ensemble for both experiments, but we see that the spread is remarkably reduced when using relaxation.

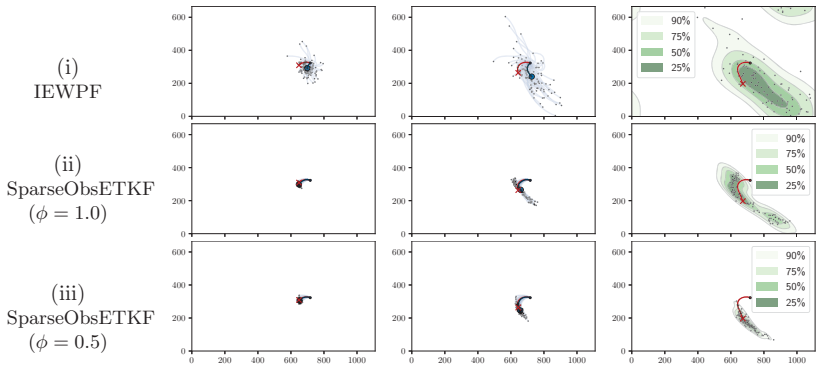
In general, we see that even though IEWPF is able to give a good state estimation through the mean, the spread in the underlying ocean state is too large to facilitate precise drift trajectory forecasts. Furthermore, LETKF without relaxation shows clear signs of overfitting, as the forecasts have low spread and do not match the ground truth. Introducing relaxation into the SparseObsETKF reduces this overfitting such that the true trajectories are correctly forecasted and uncertainty is better represented. Even more important, giving



(a) Drifter 1



(b) Drifter 2



(c) Drifter 3

Figure 4.3: Drift trajectory forecasts for three different starting positions. True trajectories represented in red, and for the first two days trajectories of ensemble members are light blue and the ensemble mean in dark blue. For the third day, the forecasted drifter positions of all ensemble members are shown with black dots and selected levels of the estimated kernel density are visualised.

more weight to the forecast that comes from the physical model prevents the ensemble from unintended anomalies. The drift trajectories estimation draws attention away from the ocean states towards dynamic visual characteristics in the ensemble.

4.4 Discussion of skill scores

Complementary to drift trajectory forecasts, we look into characteristics of both methods during the data assimilation phase between day 3 and day 10. We compare statistical properties of the ensemble against the observation data. In this setting, the key idea of skill scores is to evaluate how reliably the ensemble can forecast the next observation. An illustrative introduction with a lot of examples from atmospheric weather forecasting can be found in Wilks (2005, Chapter 7). Mathematically speaking, in this setting with data comparison, a score is

$$s\left(\widehat{F}^{n,f}, \boldsymbol{y}^n\right) \in \mathbb{R}, \tag{4.4}$$

which in our case quantifies some property of the empirical distribution from an ensemble forecast $\left(\boldsymbol{H}\boldsymbol{x}_e^{n,f}\right)_{e=1}^{N_e}$ against the true observation $\boldsymbol{y}^n$, meaning hu_j and hv_j for all $j = 1, \ldots, N_Y$. We consider three different skill scores to judge the performance.

Bias After asserting the calibration of the full analysis mean in Figure 4.2a, we investigate this further by evaluating the bias of the forecast as

$$s_1^n = \frac{1}{N_Y} \sum_{j=1}^{N_Y} \left[\overline{hu}_j^{n,f} - \boldsymbol{y}_{j,1}^n + \overline{hv}_j^{n,f} - \boldsymbol{y}_{j,2}^n \right]. \tag{4.5}$$

Here, $\overline{hu}$ and $\overline{hv}$ are the ensemble means. The bias discovers systematic trends off in the estimator.

Mean square error We further investigate the distance of each ensemble member individually from the data by measuring the MSE as

$$s_2^n = \frac{1}{N_e} \sum_{e=1}^{N_e} \left[\frac{1}{N_Y} \sum_{j=1}^{N_Y} |hu_{e,j}^{n,f} - \boldsymbol{y}_{j,1}^n|^2 + |hv_{e,j}^{n,f} - \boldsymbol{y}_{j,2}^n|^2 \right]. \tag{4.6}$$

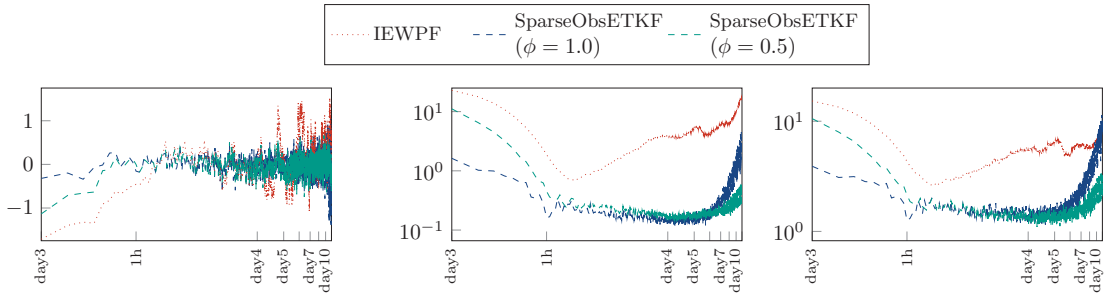
The MSE equals zero only when all ensemble members predict the observation exactly. However, this is of course not desired from a probabilistic forecast representing associated uncertainty. Nevertheless, a small MSE is desired and yields accurate fit to the data respecting the standard deviation in the observation error.

Continuous ranked probability score Similar to the integrated quadratic differences which compared distribution forecasts, we use a scoring rule that analyses the distribution of the ensemble members with the observation, see Gneiting and Raftery (2007). The CRPS is here defined by

$$s_3^n = \frac{1}{N_Y} \sum_{j=1}^{N_Y} \left[\frac{1}{N_e} \sum_{e=1}^{N_e} |hu_{e,j}^{n,f} - \boldsymbol{y}_{j,1}^n| + |hv_{e,j}^{n,f} - \boldsymbol{y}_{j,2}^n| - \frac{1}{2{N_e}^2} \sum_{e=1}^{N_e} \sum_{k=1}^{N_e} |hu_{e,j}^{n,f} - hu_{k,j}^{n,f}| + |hv_{e,j}^{n,f} - hv_{k,j}^{n,f}| \right]. \tag{4.7}$$

Large CRPS values can originate from bias (first terms) or the spread in the ensemble (last terms). Together with the scores for the bias and MSE, this allows one to identify the source of ensemble errors and to infer the properties of the ensemble.

Figure 4.4 presents the evolution of these skill scores for each data assimilation time step. These results are obtained from the same run as in Section 4.3. Note that when we assimilate the first observation after spin-up on day 3, the spread in the ensemble is relatively large by construction for all methods. It should be noted that hu and hv take values up to $500\,\mathrm{m}^2/\mathrm{s}$, which means that all methods have a relatively small



(a) Bias (b) MSE (c) CRPS

Figure 4.4: Evolution of skill scores for the IEWPF (red), SparseObsETKF without relaxation (blue), and SparseObsETKF with relaxation (turquoise) for the data assimilation phase in the experiment.

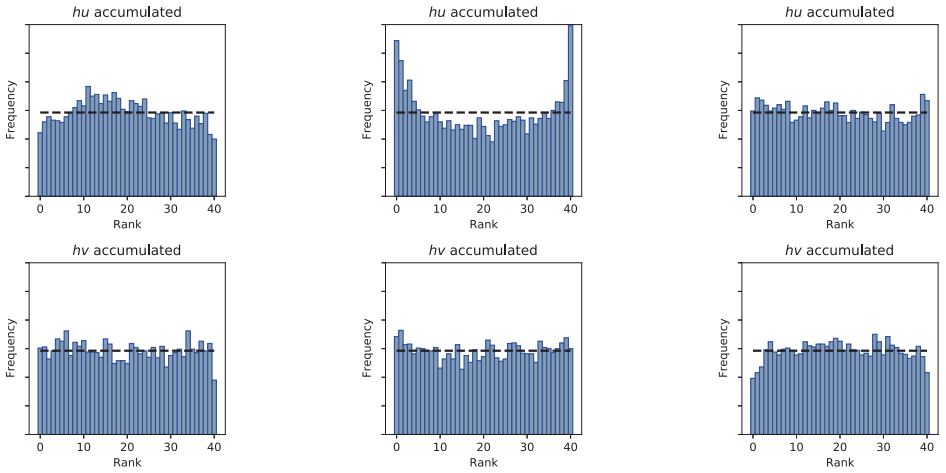
bias. In the starting phase, the SparseObsETKF immediately calibrates to the observations, whereas the IEWPF and relaxed SparseObsETKF require several data assimilation steps to correct the bias. We see however, that the bias for all methods grow over time, but with the relaxed SparseObsETKF keeping the smallest values. The systematic bias from the relaxation becomes negligible as it is sufficiently often reduced by repeated weighting with the unbiased analysis.

Looking at the MSE and CRPS, we see that both SparseObsETKF versions improve during the first few assimilation steps and stabilise at a certain level. As expected, the initial improvement with relaxation is slower than without relaxation, but this gap is closed already after 1 simulation hour, which corresponds to 12 data assimilation cycles. The quality of both SparseObsETKF versions are then stable until approximately simulation day 5, when the model dynamics gets more turbulent. At this point, the solution without relaxation starts to deteriorate due to the overfitting. Note that we see a similar trend for the relaxed SparseObsETKF later in the experiment. By relaxing even more ($\phi = 0.25$, not shown), we confirm the trend with even slower convergence in the beginning and later divergence at the end.

Similarly to the relaxed SparseObsETKF, IEWPF also converges during the initial data assimilation cycles, but the skill scores do not stabilise and instead diverge slowly. The slow initial convergence was also pointed out in Section 3, where we had to run the data assimilation sufficiently long to reach a stable level before being able to provide a fair comparison.

Note that the ensemble variance can be derived from MSE and bias. Then we see that the forecast variance behaves qualitatively similar as the MSE (not shown here). For a full assessment of the skills of a data assimilation method a single skill score gives only limited information. But for instance, the combinations of bias and CRPS broadens the insights, since the bias helps to explain the contributions in the CRPS. However, the differences especially between SparseObsETKF without relaxation and IEWPF in the skill score results do not seem substantial, whereas we have seen contrary properties in the drift trajectories that stay concealed in the monitoring of the skill scores. In general, this discussion tells us that the SparseObsETKF assimilates the ensemble much stronger towards data than the IEWPF and exemplifies the effects of relaxation.

Rank histograms We next look at rank histograms to analyse the adequacy of the ensemble spread. A short time-span in the simulation is repeated multiple times and the rank of the simulation truth in the ensemble ordering is monitored at six dynamically independent locations. Rank histograms then present the frequency of which a certain rank is reported among the N_e realisations of the ensemble and the shape of the histograms is used as a diagnostic tool to identify shortcomings of methods (Saetra et al., 2004). Flat rank histograms are commonly understood as indication for ensemble consistency or reliability of the ensemble, as it means that every ensemble member is sampled from the same distribution as the truth.



(a) IEWPF (b) SparseObsETKF ($\phi = 1.0$) (c) SparseObsETKF ($\phi = 0.5$)

Figure 4.5: Rank histograms recording the rank of the true observation within the ensemble for the observed variables. The dashed line indicates the hypothetical uniform distribution.

In Figure 4.5, we show rank histograms from repeating our experiment 1000 times, using $N_e = 40$ and simulating the first hour of data assimilation after the spin-up only. The most striking result is the clear U-shape in hu for SparseObsETKF without relaxation, which indicates that the truth often is an outlier in the ensemble and that the ensemble is underdispersive. Furthermore, we observe that IEWPF produces a slight hill-shaped rank histogram for hu, corresponding to an overdispersive ensemble. Both these observations match well with what we saw in Section 4.3. In comparison, the hu rank histogram for the relaxed SparseObsETKF closely resembles a uniform distribution. Note also that the rank histograms for hv are flatter for all three methods, but with a slight tendency towards overdispersion for the relaxed SparseObsETKF. This might be from the nature of the problem, as almost all dynamics in the case is along the x-axis.

While the rank histograms give insights how able the ensemble is to respect the uncertainty and we are able to draw similar assertions from them as we suspected already before, Hamill (2001) and Wilks (2011) advice to be careful with their interpretation, since, e.g. spatial effects between the different locations become hidden.

Summary Based on these results for the nonlinear model, we see that the CRPS together with the bias are a good start for an analysis of the ensembles during the data assimilation phase. These scalar scores are simple to include into any data assimilation sequence. By also analysing statistics over all state variables, we are able to identify additional spatial artefacts and a fundamentally different variance in the ensemble. Even though the rank histogram for the IEWPF looks reasonably flat and we get a well-calibrated mean for the state estimation, we see through the standard deviation that there are artefacts in the ensemble, leading to a higher spread than what we see for SparseObsETKF. This also made us realise how sensitive IEWPF is to the covariance structure in the model error.

Stress-testing the SparseObsETKF in this high-dimensional non-linear experiments with very sparse data discloses that the SparseObsETKF has a tendency of overfitting to the observations, resulting in an underestimated variance in the ensemble. SparseObsETKF is also not able to correctly estimate the unobserved variable η. In practice, we see that relaxing the ensemble to prior perturbations is a good

remedy for these flaws. It significantly weakens the defects, but still inherits the structures from the full SparseObsETKF. We experienced that it is not necessary to fine-tune the relaxation parameter as results were similar for $\phi = 0.25$ and $\phi = 0.75$. We further point out that we tested classical covariance inflation, but this led to nonphysical states for η, while the variance of hu and hv was barely effected. Hence, it is fair to use the SparseObsETKF and the in-built relaxation with caution.

5 Conclusion

We have compared two conceptually different state-of-the-art ensemble-based data assimilation methods, namely IEWPF and ETKF with localisation, with emphasis on the handling of sparse observation data and studied how their performances compare to each other. We have shown how localised updates are implemented by design in the IEWPF, provided that the model error covariance matrix has a local pattern, and that observations are only spread in a certain radius around the observation locations and that this is especially enhanced for sparse observations. For the ETKF, we employ a explicit localisation scheme that gives good control of computations in reduced dimensions and of the correlations between observations. We have considered two distinct cases, both motivated by simplified models applicable to oceanography. The first case studied state estimation of a linear Gaussian advection diffusion model, for which we also computed the analytical filtering distribution. This facilitated for an in-depth statistical verification of the two methods in terms of estimation of the mean, covariances, distribution coverage, spectral radius and spatial-temporal connectivity. In the assessment, which also included the standard ETKF, we recorded the performance of the ensemble-based methods in relation to the number of ensemble members and observation size. The second case was a non-linear shallow water model used for forecasting of drift trajectories. Here, we compared the performance of SparseObsETKF and IEWPF in terms of skill scores and forecast abilities. We also discussed relaxation for the SparseObsETKF localisation scheme for this case. The extensive collections of comparison metrics allowed us to analyse several properties in the ensemble representations.

Our results for the first case verified that both the IEWPF and the SparseObsETKF give very good estimates of the analytical reference solution. Additionally, we exhibit effects of different localisation parametrisations for the SparseObsETKF and IEWPF. For moderate ensemble sizes, both methods delivered on par with the KF and clearly outperformed the ETKF in terms of RMSE and coverage probabilities. ETKF was best at estimating the covariance matrices, but it suffers from spurious correlations in the updates. The SparseObsETKF yields small divergences independent of the ensemble size. In the estimation of spatio-temporal model correlations, our results revealed that all three methods performed quite evenly. ETKF converges fastest with respect to the ensemble size. Interestingly, we found that SparseObsETKF and IEWPF only showed minor improvements when increasing the ensemble size, meaning that it is most beneficial to choose one of these methods when computational resources are limited. IEWPF was the scheme benefiting the most from increased number of observations.

In the non-linear case, we learned that both IEWPF and SparseObsETKF gave estimations of the observed momentum variables with bounded errors, but SparseObsETKF without relaxation did so at the expense of nonphysical fields and small spread, causing the drift trajectory forecast in some cases to diverge from the truth. These issues were also seen in the skill scores. IEWPF, on the other hand, showed artefacts around observation sites, indicating that the model error correlation matrix might not always represent the optimal mapping for assimilating the observations. This also caused a large spread in the forecasted drift trajectories. In the case of SparseObsETKF, we showed that applying relaxation clearly improved calibration, resulting in very good general performance. These results were backed up with high-quality results in the skill scores throughout the data assimilation period and precise predictions of the drift trajectories.

To summarise, the most important findings in this paper where we consider spatially very sparse observations can be listed as follows:

- Evaluating a broad range of statistical metrics and skill scores proves to be a huge advantage as it gives a deep insight into the data assimilation methods that are not obtainable through only looking at single metrics. We recommend to start with testing bias and CRPS as one-dimensional quantities, but urge to continue with qualitative analysis for the entire spatial field.

- We strengthen the argument that IEWPF, in contrast to most other PFs, is applicable to high-dimensional applications, but that its results are highly dependent on the structure of model error covariance matrix.
- We have seen that the SparseObsETKF works well even for relatively small ensemble sizes, but good calibration of the relaxation is required to retain good results also for complex models. Upon proper relaxation, the results from the SparseObsETKF outperformed the ones of the IEWPF.

These results moreover open up new directions for future research. For instance, it would be interesting to investigate in more detail how sensitive IEWPF is to the structure of the model error covariance matrix. In our work, we tested our implementation of the covariance localisation only with respect to the ETKF. It would be interesting to check whether other variants of the EnKF work equally well, or if they have advantages or disadvantages over ETKF. Beyond this, the sensitivity of the localisation concept to the relaxation parametrisation could be tuned adaptively. Finally, it would also be interesting to test the IEWPF and the SparseObsETKF in a real-world settingby assimilating real observations into an ensemble of simplified ocean models with the aim to predict true drift trajectories.

Code Availability

The source code and data used to produce the results presented in this paper are openly available under a GPL3 licence at `http://doi.org/10.5281/zenodo.10039791` for Section 3 and at `http://doi.org/10.5281/zenodo.10037281,http://doi.org/10.5281/zenodo.10039872` for Section 4.

Acknowledgement

The authors would like to thank Kjetil Olsen Lye for valuable feedback on the manuscript. Furthermore, we thank the two anonymous reviewers for their comments that improved the manuscript further.

References

Anderson, J. L. (2009). Spatially and temporally varying adaptive covariance inflation for ensemble filters. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 61 A(1):72–83.

Anderson, J. L. and Anderson, S. L. (1999). A Monte Carlo implementation of the nonlinear filtering problem to produce ensemble assimilations and forecasts. *Monthly Weather Review*, 127(12):2741–2758.

Asch, M., Bocquet, M., and Nodet, M. (2016). *Data assimilation: methods, algorithms, and applications*. SIAM.

Bishop, C. H., Etherton, B. J., and Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform kalman filter. part i: Theoretical aspects. *Monthly Weather Review*, 129(3):420–436.

Breivik, Ø., Allen, A. A., Maisondieu, C., and Olagnon, M. (2013). Advances in search and rescue at sea. *Ocean Dynamics*, 63(1):83–88.

Brodtkorb, A. R. and Holm, H. H. (2021). Coastal ocean forecasting on the GPU using a two-dimensional finite-volume scheme. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 73(1):1–22.

Burgers, G., Van Leeuwen, P. J., and Evensen, G. (1998). Analysis scheme in the ensemble Kalman filter. *Monthly Weather Review*, 126(6):1719–1724.

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G. (2018). Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5):1–50.

Chertock, A., Dudzinski, M., Kurganov, A., and Lukáčová-Medvid'ová, M. (2018). Well-balanced schemes for the shallow water equations with Coriolis forces. *Numerische Mathematik*, 138(4):939–973.

Chopin, N. and Papaspiliopoulos, O. (2020). *An Introduction to Sequential Monte Carlo Methods*. Springer.

Desroziers, G., Berre, L., Chapnik, B., and Poli, P. (2006). Diagnosis of observation, background and analysis-error statistics in observation space. *Quarterly Journal of the Royal Meteorological Society*, 131(613):3385–3396.

Doucet, A., Godsill, S., and Andrieu, C. (2000). Methods for Bayesian filtering. *Statistics and Computing*, pages 197–208.

Evensen, G. (1994). Sequential data assimilation with a nonlinear quasi-geostrophic model using Monte Carlo methods to forecast error statistics. *Journal of Geophysical Research*, 99(C5):10143–10162.

Evensen, G. (2003). The Ensemble Kalman Filter: Theoretical formulation and practical implementation. *Ocean Dynamics*, 53(4):343–367.

Evensen, G. (2009). *Data Assimilation. The Ensemble Kalman Filter*. Springer.

Foss, K. H., Berget, G. E., and Eidsvik, J. (2021). Using an autonomous underwater vehicle with onboard stochastic advection-diffusion models to map excursion sets of environmental variables. *Environmetrics*, 33(November 2020):1–18.

Gaspari, G. and Cohn, S. E. (1999). Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757.

Gneiting, T. and Raftery, A. E. (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378.

Greybush, S. J., Kalnay, E., Miyoshi, T., Ide, K., and Hunt, B. R. (2011). Balance and ensemble Kalman filter localization techniques. *Monthly Weather Review*, 139(2):511–522.

Hamill, T. M. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560.

Holm, H. H. (2020). *Efficient Forecasting of Drift Trajectories using Simplified Ocean Models and Nonlinear Data Assimilation on GPUs*. PhD thesis, Norwegian University of Science and Technology.

Holm, H. H., Sætra, M. L., and van Leeuwen, P. J. (2020). Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting. *Journal of Computational Physics: X*, 6(0314):100053.

Houtekamer, P. L. and Mitchell, H. L. (1998). Data assimilation using an ensemble kalman filter technique. *Monthly Weather Review*, 126(3):796–811.

Houtekamer, P. L. and Mitchell, H. L. (2001). A sequential ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 129(1):123–137.

Houtekamer, P. L. and Zhang, F. (2016). Review of the ensemble Kalman filter for atmospheric data assimilation. *Monthly Weather Review*, 144(12):4489–4532.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126.

Kirchgessner, P., Nerger, L., and Bunse-Gerstner, A. (2014). On the choice of an optimal localization radius in ensemble Kalman filter methods. *Monthly Weather Review*, 142(6):2165–2175.

Li, H. (2007). *Local Ensemble Transform Kalman Filter*. PhD thesis, University of Maryland.

Li, H., Kalnay, E., and Miyoshi, T. (2009). Simultaneous estimation of covariance inflation and observation errors within an ensemble Kalman filter. *Quarterly Journal of the Royal Meteorological Society*, 135(639):523–533.

Morzfeld, M., Hodyss, D., and Snyder, C. (2017). What the collapse of the ensemble Kalman filter tells us about particle filters. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 69(1):1–15.

Nerger, L. (2015). On serial observation processing in localized ensemble Kalman filters. *Monthly Weather Review*, 143(5):1554–1567.

Nussbaumer, R., Mariethoz, G., Gloaguen, E., and Holliger, K. (2018). Which Path to Choose in Sequential Gaussian Simulation. *Mathematical Geosciences*, 50(1):97–120.

Ott, E., Hunt, B. R., Szunyogh, I., Zimin, A. V., Kostelich, E. J., Corazza, M., Kalnay, E., Patil, D., and Yorke, J. A. (2004). A local ensemble Kalman filter for atmospheric data assimilation. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):415–428.

Raanes, P. N., Bocquet, M., and Carrassi, A. (2019). Adaptive covariance inflation in the ensemble Kalman filter by Gaussian scale mixtures. *Quarterly Journal of the Royal Meteorological Society*, 145(718):53–75.

Röhrs, J., Dagestad, K. F., Asbjørnsen, H., Nordam, T., Skancke, J., Jones, C. E., and Brekke, C. (2018). The effect of vertical mixing on the horizontal drift of oil spills. *Ocean Science*, 14(6):1581–1601.

Saetra, O., Hersbach, H., Bidlot, J. R., and Richardson, D. S. (2004). Effects of observation errors on the statistics for ensemble spread and reliability. *Monthly Weather Review*, 132(6):1487–1501.

Sætrom, J. and Omre, H. (2013). Uncertainty quantification in the Ensemble Kalman filter. *Scandinavian Journal of Statistics*, 40(4):868–885.

Sakov, P. and Bertino, L. (2011). Relation between two common localisation methods for the EnKF. *Computational Geosciences*, 15(2):225–237.

Scott, D. W. (1992). *Multivariate density estimation: Theory, practice, and visualization: Second edition*. Wiley.

Sigrist, F., Künsch, H. R., and Stahel, W. A. (2015). Stochastic partial differential equation based modelling of large space-time data sets. *Journal of the Royal Statistical Society. Series B: Statistical Methodology*, 77(1):3–33.

Skauvold, J., Eidsvik, J., van Leeuwen, P. J., and Amezcua, J. (2019). A revised implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 145(721):1490–1502.

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review*, 136(12):4629–4640.

Soares, R. V., Luo, X., Evensen, G., and Bhakta, T. (2021). Handling big models and big data sets in history-matching problems through an adaptive local analysis scheme. *SPE Journal*, 26(2):973–992.

Szunyogh, I., Satterfield, E. A., Elana, J., Gyarmati, G., Kalnay, E., Hunt, B. R., Eric, J., Kuhl, D. D., Ott, E., and Yorke, J. A. (2007). The Local Ensemble Transform Kalman Filter and its implementation on the NCEP global model at the University of Maryland The LETKF and its implementation on the NCEP GFS model. *Proc. ECMWF Workshop on flow-dependent aspects of data assimilation*, pages 1–18.

Thorarinsdottir, T. L., Gneiting, T., and Gissibl, N. (2013). Using proper divergence functions to evaluate climate models. *SIAM-ASA Journal on Uncertainty Quantification*, 1(1):522–534.

Van Leeuwen, P. J. (2009). Particle filtering in geophysical systems. *Monthly Weather Review*, 137(12):4089–4114.

van Leeuwen, P. J. (2020). A Consistent Interpretation of the Stochastic Version of the Ensemble Kalman Filter. *Quarterly Journal of the Royal Meteorological Society*, 146(731):2815–2825.

van Leeuwen, P. J., Künsch, H. R., Nerger, L., Potthast, R., and Reich, S. (2019). Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723):2335–2365.

Vetra-Carvalho, S., van Leeuwen, P. J., Nerger, L., Barth, A., Altaf, M. U., Brasseur, P., Kirchgessner, P., and Beckers, J. M. (2018). State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 70(1):1–38.

Whitaker, J. S. and Hamill, T. M. (2002). Ensemble data assimilation without perturbed observations. *Monthly Weather Review*, 130(7):1913–1924.

Wilks, D. S. (2005). *Statistical methods in the atmospheric sciences*. Elsevier Inc.

Wilks, D. S. (2011). On the reliability of the rank histogram. *Monthly Weather Review*, 139(1):311–316.

Zhang, F., Snyder, C., and Sun, J. (2004). Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble kalman filter. *Monthly Weather Review*, 132(5):1238–1253.

Zhu, M., van Leeuwen, P. J., and Amezcua, J. (2016). Implicit equal-weights particle filter. *Quarterly Journal of the Royal Meteorological Society*, 142(698):1904–1919.

Paper IV

Multi-Level Data Assimilation for Simplified Ocean Models

Florian Beiser, Håvard Heitlo Holm, Kjetil Olsen Lye, Jo Eidsvik

In review

Multi-level Data Assimilation for Simplified Ocean Models

Florian Beiser[1,2], Håvard Heitlo Holm[1], Kjetil Olsen Lye[1], and Jo Eidsvik[2]

[1]Mathematics and Cybernetics, SINTEF Digital, Oslo, Norway
[2]Department of Mathematical Sciences, NTNU, Trondheim, Norway

Abstract

Multi-level Monte Carlo methods have become an established technique in uncertainty quantification as they provide the same statistical accuracy as traditional Monte Carlo methods but with increased computational performance. Recently, similar techniques using multi-level ensembles have been applied to data assimilation problems. In this work we study the practical challenges and opportunities of applying multi-level methods to complex data assimilation problems, in the context of simplified ocean models. We simulate a simplified ocean model, based on the shallow-water equations, at different resolutions and employ a multi-level Kalman filter to assimilate sparse in-situ observations. In this context, we present numerical results from a synthetic test case, where small-scale perturbations lead to turbulent behaviour, conduct state estimation and forecast drift trajectories using multi-level ensembles. This represents a new advance towards making multi-level data assimilation feasible for real-world oceanographic applications.

1 Introduction

Within atmospheric and oceanographic applications, it is common to represent the uncertainty in a forecast through Monte Carlo based ensemble simulations. Here different realisations of a numerical prediction are created by sampling uncertain factors, such as initial conditions and forcing, and the spread in the resulting equiprobable forecasts is used to estimate the uncertainty. Commonly, all ensemble members have the same level of accuracy, meaning that they rely on the same physical model, numerical method, and spatial discretisation. The use of such ensembles can therefore be termed a single-level Monte Carlo experiment. While the single-level Monte Carlo framework is generally well-functioning, convergence to the true statistics is slow as the ensemble size increases, leading to long computational run times and high associated costs.

Multi-level Monte Carlo (MLMC) methods have become an established tool in the realm of uncertainty quantification, primarily due to their computational efficiency (Giles, 2015). MLMC methods utilise a hierarchy of ensembles defined on different levels of accuracy to estimate forecast uncertainty and other statistical quantities (Giles, 2008). In spatio-temporal applications, the different levels often correspond to simulating the same physical model with the same numerical method but with varying grid resolutions. With lower resolution simulations being more computationally efficient (Mishra and Schwab, 2012). In addition to a classical Monte Carlo ensemble on the coarsest resolution, multi-level ensembles consist of coupled ensembles between the finer resolutions. The underlying idea of this hierarchical structure is that levels associated with low accuracy are computationally inexpensive, allowing for large ensemble sizes and thereby achieving reliable statistical estimates of low-level features. Higher-level details in the forecast are then estimated from structures appearing in the coupled ensembles at increasingly higher levels. Under the assumption that the numerical solutions of the same stochastic realisation become more and more similar on higher levels, the ensemble sizes of levels associated with higher accuracy can be reduced. As a result, MLMC estimators can achieve the same theoretical, statistical accuracy as classical single-level estimators but at reduced computational cost (Giles, 2015).

To define relevant initial conditions for an ensemble forecast, it is essential to make use of recent available observations of the true state of the system in question. Observations are often diverse and noisy, and are

rarely available for all state variables in the forecast model. They can also be indirect, meaning that the state variables themselves are not observed directly. It is therefore necessary to combine information from the prior numerical forecast with available observations. The natural framework for doing this is through data assimilation (e.g., Evensen et al. (2022)). Particularly, for forecasting systems based on single-level ensembles, ensemble Kalman filters (EnKF) represent a widely popular class of sequential data assimilation methods (Carrassi et al., 2018). To leverage MLMC for real-world geophysical problems, it is beneficial to extend the multi-level framework to also include data assimilation. We refer to this as multi-level data assimilation (MLDA). Even though MLDA is strictly speaking a subset of MLMC, we distinguish between the two by using MLMC when the multi-level ensemble is not conditioned on observations, and MLDA when it is framed in the context of data assimilation. Work on MLDA has largely concentrated on theoretical analyses (Hoel et al., 2016; Beskos et al., 2017; Jasra et al., 2020; Chernov et al., 2021). In this paper, we explore the applicability of MLDA to geophysical problems with real-world relevance.

Our motivation to exploit MLDA arises from search-and-rescue operations at sea, where efficient prediction of drift trajectories with associated uncertainties is of utmost importance. Drift trajectory models used operationally (e.g., Dagestad et al. (2018)) typically compute the trajectories using available forecasts of the ocean currents, generated by operational circulation models (e.g., Shchepetkin and McWilliams (2005)). Operational circulation models are computationally expensive, and they have to compete for computational resources with other meteorological and oceanographic operational models. As a result, available input fields for the trajectory model are often deterministic or from very small ensemble sizes, and although uncertainty can be introduced through parameter sensitivity related to the drifting object and its initial position, uncertainties in the underlying ocean currents are not properly accounted for. To complement this approach, Røed (2012) suggested running large ensembles of computationally efficient simplified ocean models to quantify the uncertainty in the ocean currents, and consequently also the uncertainty in drift trajectories. With this approach, it also becomes feasible to assimilate in-situ observations of ocean currents into the ensemble of simplified models. Such observations can be obtained from local buoys or drifters released during search-and-rescue operations (Rabault et al., 2022) and can provide important information to improve numerical estimates of local conditions. However, buoy and drifter observations are sparse, measurements are located far apart from each other and there are few of them relative to the size of the state space.

Within this context, we aim to investigate the applicability of a recently proposed MLDA method incorporated into an existing simulation framework for running computationally efficient, single-level ensembles of simplified ocean models (Brodtkorb and Holm, 2021). The simulation framework uses finite-volume schemes, and has previously been used to study data assimilation methods for sparse observations relevant to search-and-rescue (Holm et al., 2020a,b).

The key research questions in this work are:

i) Given an existing simulation framework for computationally efficient single-level ensemble simulations, how can we leverage MLMC methods to improve computational efficiency? Additionally, what performance gains can MLDA deliver if the simulation framework is not optimised for simulating on very coarse computational grids?

ii) Recognising that the practical use of MLDA wants to go beyond the common MLMC tuning that has been focused on a single quantity of interest, how can we configure fixed multi-level ensemble sizes for MLDA? That is, how many realisations to use at each level?

iii) Multi-level ensembles for MLDA can be configured with respect to different quantities of interest. How does the configuration of the multi-level ensemble influence the quality of the data assimilation? And how does the posterior distribution from MLDA experiments compare to equivalent similar single-level data assimilation?

iv) Keeping a focus on predictive properties, how can the posterior approximation from a multi-level ensemble be used for drift trajectory forecasting that is consistent with single-level forecasts?

1.1 Related work

The main scientific developments in this work on MLDA build on the fields of MLMC methods and ensemble-based data assimilation. We therefore present a brief overview of relevant work in these domains.

MLMC methods efficiently approximate quantities of interest that involve high-dimensional integrals. The approach was originally introduced for numerical integration (Heinrich, 2001). In the field of uncertainty quantification, the method was first presented for the estimation of expected values (Giles, 2008). The multi-level estimator was constructed by using approximations of increasing accuracy, choosing ensembles on the different levels, and employing a telescoping sum (Giles, 2015). For this concept, Schaden and Ullmann (2020) first derived ways to construct the best linear unbiased estimators in the scalar case with fixed computational budget, before Destouches et al. (2023) did the same in the multi-dimensional case. With numerical solutions to partial differential equations (PDEs), the hierarchy of levels was constructed by solving the problem on increasingly fine grids, and using the MLMC estimators on the results (Dodwell et al., 2015). The cases of hyperbolic conservation laws and finite volume methods have been separately analysed for the application of MLMC by Mishra and Schwab (2012) and Fjordholm et al. (2020), including the particular model of the shallow-water equations (Mishra et al., 2012b).

Ensemble-based data assimilation refers to a set of techniques that assimilate observations into a probabilistic forecast generated by running an ensemble of numerical models (Evensen et al., 2022). The most common methods can be subdivided into ensemble variational methods (EnVar), particle filters (PF), and ensemble Kalman filters (EnKF), and tutorial presentations of these methods with emphasis on geophysical problems can be found in e.g. Carrassi et al. (2018) and Vetra-Carvalho et al. (2018). Plenty adaptions to the characteristics of the model or the data exist, including approaches for drifter observations (Sun and Penny, 2019). Variational methods solve a numerical optimisation problem that seeks to minimise the discrepancy between the numerical model and the observations with respect to the observation uncertainties, and the difference between the prior state and the posterior with respect to the so-called background error covariances and model errors (Bannister, 2017). EnVar seeks to estimate the background error covariance using an ensemble. In contrast, PFs are non-linear filters that in their standard form evaluate Bayes' rule directly by adjusting the relative importance or weights of each ensemble member according to its resemblance to the observations (Van Leeuwen et al., 2019). EnKF methods apply an assimilation step which is derived from the linear Gaussian model, but this method still proves highly useful in many non-linear applications and it is therefore popular in practice (Evensen, 2006).

Most of the mentioned approaches for data assimilation have been tested in the setting of MLMC. Among the multi-level version of PFs, the weights are updated first on the coarsest level as in the single-level PF, and then on increasingly finer coupled levels (Jasra et al., 2017; Latz et al., 2018). Bierig and Chernov (2014) and Mycek and De Lozzo (2019) phrased EnVar in the context of multi-level ensembles. The approach of multi-level EnKF (MLEnKF) was first introduced for temporal processes by Hoel et al. (2016), by combining the EnKF and MLMC frameworks through the use of multi-level ensembles and multi-level estimators for the approximation of the Kalman gain. This result was extended to spatio-temporal processes by Chernov et al. (2021), and similar approaches were developed in history matching where the multi-level ensembles are complemented with hybrid filters (Fossum et al., 2020; Nezhadali et al., 2020).

1.2 Contribution and outline

In this work, we discuss and assess the applicability of MLDA techniques for practical ocean forecasting applications. We focus on the simplified ocean models represented by rotating shallow-water equations, assimilating observations in a challenging test case with turbulent behaviour. We develop an approach using the posterior multi-level ensemble to forecast drift trajectories. Our numerical results show that we obtain the same statistical quality for state estimation as we do using a single-level EnKF with an ensemble size corresponding to the same theoretical error, but with a computational speed-up of up to a factor two. This work advances MLDA one step further, from theoretical proof-of-concept studies towards practical oceanographic problems.

Our focus in this paper is EnKF data assimilation methods and MLMC. We revise the MLEnKF method

introduced by Chernov et al. (2021) to a finite-volume framework and the common situation of sparse observations. We direct attention to a set of emerging practical challenges, including the choice of multi-level ensemble sizes, incorporation of localisation strategies, and mitigation of negative eigenvalues in covariance estimations which is a known challenge associated with multi-level methods. Hereby, we discuss how the multi-level configuration for MLDA is influenced by the scaling of computational costs and different quantities of interest, and how these factors influence the potential for computational speed up when using multi-level ensembles. Furthermore, we use an existing simulation framework for shallow-water simulations, which is optimised for high computational performance on fine spatial resolution.

The rest of the paper is structured as follows. In Sect. 2, we repeat the basis of Monte Carlo methods with emphasis on multi-level statistics. Then, we introduce the general data assimilation problem and describe the MLEnKF in the context of numerical models based on finite-volume methods in Sect. 3. In Sect. 4, we elaborate on practicalities in the implementation of MLDA. In Sect. 5, we introduce the identical twin experiments using a turbulent synthetic test case initialised by two, initially well-balanced but unstable, geostrophic jets, and assimilate sparse observations into a multi-level ensemble using MLEnKF. Using this experiment, we compare numerical results for single-level EnKF and MLEnKF. Sect. 6 is dedicated to forecasting of drift trajectories, where we demonstrate practical use of the obtained posterior multi-level ensembles by showing how to use them for drift trajectory modelling and forecasting. In Sect. 7, we finish with some concluding remarks.

2 Single- and multi-level Monte Carlo methods

For a comprehensive presentation of the underlying Monte Carlo methods, we start by recapping the single-level Monte Carlo approach in Sect. 2.1 and the MLMC approach in Sect. 2.2.

We consider a random vector $\boldsymbol{x} \in \mathbb{R}^{n_x}$ which will be generated by a model $\mathcal{M}$. For spatio-temporal applications of our interest, $\mathcal{M}$ represents a simplified ocean model simulated numerically by a finite-volume method and $\boldsymbol{x}$ is defined on a grid and its entries represent grid cell averages of physical variables for all grid cells in a discretised domain at a certain time t. This means that the size n_x of the state vector equals the product of the number of grid cells in the domain and the number of physical variables included in the model.

2.1 Single-level Monte Carlo

To represent the probability distributions for the state, we use ensembles $(\boldsymbol{x}_e)_{e=1}^N$, where N is the ensemble size. All ensemble members share the same state dimension, and we refer to this approach as the *single-level* method. We use Monte Carlo estimation to infer statistical quantities of interest, with notation

$$\overline{\boldsymbol{x}} = \frac{1}{N}\sum_{e=1}^{N} \boldsymbol{x}_e \tag{1a}$$

and

$$\tilde{\boldsymbol{x}}_e = \boldsymbol{x}_e - \overline{\boldsymbol{x}} \tag{1b}$$

for the ensemble average and perturbations, respectively. Then, the unbiased single-level estimators for the mean and the error covariance matrix are

$$\boldsymbol{\mu}^{\mathrm{SL}}[\boldsymbol{x}] = \overline{\boldsymbol{x}} \tag{2a}$$

and

$$\boldsymbol{\Sigma}^{\mathrm{SL}}[\boldsymbol{x}, \boldsymbol{x}] = \frac{1}{N-1}\sum_{e=1}^{N} \tilde{\boldsymbol{x}}_e (\tilde{\boldsymbol{x}}_e)^\top, \tag{2b}$$

where we use the superscript SL to indicate single-level estimators, and where $\top$ denotes transpose. More generally, for a quantity of interest represented by the function $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_g}$, the single-level Monte Carlo

estimator approximates

$$\mathbb{E}\left(g(\boldsymbol{x})\right) \approx \frac{1}{N}\sum_{e=1}^{N} g(\boldsymbol{x}_e). \tag{3}$$

In terms of computer resources, there is a trade off between the grid resolution and the ensemble size. If c denotes the computational cost of generating a sample $\boldsymbol{x}$, the total cost of a single-level ensemble is

$$C^{\mathrm{SL}} = Nc. \tag{4}$$

The statistical accuracy τ^2 is the variance of the estimator given by

$$\tau^2 = \frac{\|\mathbb{V}\mathrm{ar}[g(\boldsymbol{x})]\|_2}{N}, \tag{5}$$

where $\|\cdot\|_2$ denotes the L^2-norm. For unbiased estimators, this can be interpreted as the expected square sampling error of the estimator with respect to $\mathbb{E}\left(g(\boldsymbol{x})\right)$. For fixed grid resolution, the statistical accuracy depends on the number of ensemble members and the quantity of interest.

2.2 Multi-level Monte Carlo

While we do not aim to provide a full review of the MLMC approach, we recall the basics for the sake of completeness and for introducing consistent notation. In contrast to single-level Monte Carlo methods, where all state vectors in an ensemble are defined on the same grid, MLMC relies on the definition of a hierarchy of grids.

2.2.1 Hierarchy of grids

Starting with the original grid of interest that has the finest resolution, we define a hierarchy of computationally less expensive grids. The levels in this hierarchy of discretisations are denoted with superscripts $0, \ldots, L$, where L is the finest level. The corresponding state vectors are represented by $\boldsymbol{x}^l \in \mathbb{R}^{n_x^l}$ for levels $l = 0, \ldots, L$ with corresponding dimensions $n_x^0 < \cdots < n_x^L$, where $n_x^L = n_x$ for consistency with the single-level notation.

To project state vectors and other quantities from one level to another, it is convenient and computationally efficient to use nested grids, see, e.g. Mishra and Schwab (2012). For a two-dimensional Cartesian grid for a finite-volume scheme, for example, we can easily define nested grids by iteratively merging a set of neighbouring cells from one level to the coarser level, ensuring that all cell faces on level $l-1$ match that on the finer level l. Conversely, any single grid cell on level l is covered by one and only one grid cell on the coarser level $l-1$. Herein, we consider nested grids where a single grid cell on level $l-1$ corresponds to a block of two by two grid cells on level l, representing a coarsening factor of two.

For the up- and downscaling of finite-dimensional vectors between grids for levels l and $l-1$, there are several possible techniques, but it is helpful to take into consideration what the state variables actually represent (Farmer, 2002). For example, if the state vector represents point values, linear interpolation may be adequate. In our case, where we have nested grids and the state vector represents cell average values, we use simple repetition and averaging for the up- and downscaling, respectively. Note that this ensures conservation of physical variable in accordance with the finite-volume framework, while it is also computationally efficient, see, e.g. Nezhadali et al. (2020). This enables us to execute algebraic calculations for states on different levels, but for improved readability, this choice of up- and downscaling is never explicitly used in the notation.

According to the defined levels, a hierarchy of model operators $\mathcal{M}^l$, for levels $l = 0, \ldots, L$ generates samples on the corresponding grids. In the case of consistent model equations, the different model operators $\mathcal{M}^l$ represent the same numerical solver with different underlying grids according to the level definition. The computational cost of simulating the model $\mathcal{M}^l$ depends on the numerical model itself and on the chosen discretisation. In general, coarser resolutions are significantly faster to run than finer ones. For a two-dimensional grid with cell edges of size Δx, for instance, the computational effort to simulate a simplified ocean model numerically by a finite-volume method for the shallow-water equations is

$$\mathcal{O}(\Delta x^{-3}), \tag{6}$$

see, e.g., Mishra et al. (2012b). This means that doubling Δx, or equivalently four-dividing the number of grid cells (in two dimensions), reduces the computational cost eight times. (Here, one magnitude comes from the fact that the numerical solver has to take time steps proportional to Δx.)

2.2.2 Multi-level estimators

A multi-level ensemble consists of a single-level ensemble on the coarsest level and ensembles with pairs of states at subsequent levels:

$$(\boldsymbol{x}_e^0)_{e=1}^{N^0}, \quad (\boldsymbol{x}_e^{l+}, \quad \boldsymbol{x}_e^{l-})_{e=1}^{N^l}, \quad l = 1, \ldots, L, \tag{7}$$

where $N^0, \ldots, N^L$ are the ensemble sizes per level. Here, the partners $\boldsymbol{x}_e^{l+}$ and $\boldsymbol{x}_e^{l-}$ that constitute a pair on level l share the same realisation, but $\boldsymbol{x}_e^{l+} \in \mathbb{R}^{n_x^l}$ is defined on level-l grid, whereas $\boldsymbol{x}_e^{l-} \in \mathbb{R}^{n_x^{l-1}}$ is defined on level-$(l-1)$ grid. A consequence of this is that $\boldsymbol{x}^{l+}$ and $\boldsymbol{x}^{(l+1)-}$ are defined on the same grid, but belong to different levels in the multi-level ensemble, and we therefore use the notation with $+$ and $-$ in the superscript to distinguish them. For simplicity, we also let $\boldsymbol{x}^L = \boldsymbol{x}^{L+}$ as these are the only ensemble members defined on the finest grid. Note that $\boldsymbol{x}^{l+}$ is evolved by $\mathcal{M}^l$ and $\boldsymbol{x}^{l-}$ by $\mathcal{M}^{l-1}$.

A multi-level estimator utilises a telescoping sum over the levels for the estimation of quantities of interest. This construction makes it different from the typical empirical approximation of classical Monte Carlo estimators. For the expected value, we have the following telescoping sum property

$$\mathbb{E}\left(\boldsymbol{x}^L\right) = \mathbb{E}(\boldsymbol{x}^0) + \sum_{l=1}^{L} \mathbb{E}\left(\boldsymbol{x}^{l+} - \boldsymbol{x}^{l-}\right), \tag{8}$$

and in general, for a function $g : \mathbb{R}^{n_x} \to \mathbb{R}^{n_g}$ for some $n_g \in \mathbb{N}$, we have

$$\mathbb{E}\left(g(\boldsymbol{x}^L)\right) = \mathbb{E}(g(\boldsymbol{x}^0)) + \sum_{l=1}^{L} \mathbb{E}\left(g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})\right). \tag{9}$$

The above motivates the following general approximation

$$\mathbb{E}\left(g(\boldsymbol{x}^L)\right) \approx \frac{1}{N^0} \sum_{e=1}^{N^0} g(\boldsymbol{x}^0) + \sum_{l=1}^{L} \left[\frac{1}{N^l} \sum_{e=1}^{N^l} \left(g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})\right) \right], \tag{10}$$

using a multi-level ensemble of samples. Following Eq. (10), one can define multi-level estimators for the mean and covariance as

$$\boldsymbol{\mu}^{\mathrm{ML}}[\boldsymbol{x}^L] = \overline{\boldsymbol{x}}^0 + \sum_{l=1}^{L} (\overline{\boldsymbol{x}}^{l+} - \overline{\boldsymbol{x}}^{l-}) \tag{11a}$$

and

$$\boldsymbol{\Sigma}^{\mathrm{ML}}[\boldsymbol{x}^L, \boldsymbol{x}^L] = \boldsymbol{\Sigma}^0 + \sum_{l=1}^{L} (\boldsymbol{\Sigma}^{l+} - \boldsymbol{\Sigma}^{l-}), \tag{11b}$$

respectively, where $\overline{\boldsymbol{x}}^{l+}$ and $\boldsymbol{\Sigma}^{l+}$ are the single-level mean and covariance estimators on level l estimated by the ensemble $(\boldsymbol{x}_e^{l+})_{e=1}^{N^l}$ only. (Analogous notation is used for the $(\boldsymbol{x}_e^{l-})_{e=1}^{N^l}$ ensemble.) By the use of linearity, the covariance at the finest level becomes

$$\boldsymbol{\Sigma}^{\mathrm{ML}}[\boldsymbol{x}^L, \boldsymbol{x}^L] = \frac{1}{N^0 - 1} \sum_{e=1}^{N^0} \tilde{\boldsymbol{x}}_e^0 (\tilde{\boldsymbol{x}}_e^0)^\top + \sum_{l=1}^{L} \frac{1}{N^l - 1} \sum_{e=1}^{N^l} \left(\tilde{\boldsymbol{x}}_e^{l+} (\tilde{\boldsymbol{x}}_e^{l+})^\top - \tilde{\boldsymbol{x}}_e^{l-} (\tilde{\boldsymbol{x}}_e^{l-})^\top\right), \tag{12}$$

where the differences are taken between the partners on every level directly.

2.2.3 Optimal configuration of the multi-level ensemble size

For single-level ensembles, the determination of an ensemble size is relatively straightforward and is typically based on how many samples one can afford within a given computational budget. However, in the multi-level case, the choice of ensemble size becomes more involved. Often the assumption of exponentially growing ensemble sizes towards the coarsest resolution is used to guarantee desired theoretical properties (Bierig and Chernov, 2014; Chernov et al., 2021).

For the following considerations, we fix a quantity of interest meaning that we examine the approximation of $\mathbb{E}[g(\boldsymbol{x})]$ for a given function g. Under certain assumptions, a multi-level estimator yields a computational speed-up for estimating the quantity of interest while keeping a fixed statistical accuracy τ^2. Similarly to the statistical accuracy of the single-level estimator given in Eq. (5), the accuracy of the multi-level estimator is given by

$$\tau^2 = \frac{\|\mathbb{V}\mathrm{ar}[g(\boldsymbol{x}^0)]\|_2}{N^0} + \sum_{l=1}^{L} \frac{\|\mathbb{V}\mathrm{ar}[g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})]\|_2}{N^l}, \tag{13}$$

see Müller (2014). Alternatively, from the reverse perspective, the estimator can offer improved statistical accuracy while keeping the computational work fixed (Giles, 2015; Lye, 2020). For a fixed work budget, optimal weights in the construction of the multi-level estimator could be derived (Schaden and Ullmann, 2020; Destouches et al., 2023) and strategies accounting for parallelisation should be considered (Mishra et al., 2012a).

The main idea in MLMC is that if the numerical discretisation convergences in the sense that $\|g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})\|_2 \to 0$ almost surely as $l \to \infty$, then the variance of the details $\mathbb{V}\mathrm{ar}[g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})] = \mathcal{O}(\mathbb{E}(\|g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})\|_2^2))$ will be small for large l. In other words, the sampling error on the finer levels is small, in the sense that their contribution to τ^2 in Eq. (13) is small, and as a consequence we need fewer samples on the finer, more expensive, levels, see, e.g. Giles (2015). On the coarser levels we will need more samples, but the coarser levels are cheaper to compute due to Eq. (6).

Let c^l denote the cost of generating a sample $\boldsymbol{x}^l$ at level l and let C^l be the cost of $\boldsymbol{x}^{l+} - \boldsymbol{x}^{l-}$ which actually involves two simulations. By definition, we set $C^0 = c^0$. The cost is determined by the computational work to solve the PDE numerically with some software on some hardware. The theoretical costs can be specified using Eq. (6) or must be found in practice by measuring the run time of the simulation for different numerical grid sizes. Then, for the multi-level estimation, the variances of the quantity of interest are relevant and we let

$$v^l = \|\mathbb{V}\mathrm{ar}[g(\boldsymbol{x}^l)]\|_2 \tag{14a}$$

and

$$V^l = \|\mathbb{V}\mathrm{ar}[g(\boldsymbol{x}^{l+}) - g(\boldsymbol{x}^{l-})]\|_2, \tag{14b}$$

where we set $V^0 = v^0$. Notably these variances are given by the problem at hand through the choice of g and the distribution of $\boldsymbol{x}$. These variances can be estimated beforehand using experiments or theoretical considerations of convergence in the setting of the numerical discretisation used.

Further, one can efficiently allocate computational resources, where by rule-of-thumb a speed-up is achieved if

$$V^l \ll v^0 \tag{15a}$$

and

$$C^0 \ll \cdots \ll C^L. \tag{15b}$$

For the best speed up, we need strong correlations between the levels and a strong computational scaling for the different resolutions.

Given a fixed number of levels L, corresponding variances $\{V_l\}_{l=0}^L$, and computational costs $\{C_l\}_{l=0}^L$, one can derive optimal sample numbers and a theoretical speed-up against single-level Monte Carlo. By following the approach for optimal speed-up given a statistical accuracy τ^2, the optimal multi-level ensemble sizes are given by

$$N^l = \left\lceil \sqrt{\frac{V^l}{C^l}} C_\tau \right\rceil \quad \text{with } C_\tau = \frac{1}{\tau^2} \sum_{l=0}^{L} \sqrt{V^l C^l}, \tag{16}$$

see Müller (2014). Note here, that the optimal ensemble sizes depends on τ^2, the computational performance scaling, and the quantity of interest g, as the latter influences V^l. The computational cost of the resulting multi-level ensemble is then given by

$$C^{\mathrm{ML}} = \sum_{l=0}^{L} N^l C^l. \tag{17}$$

3 Single- and multi-level data assimilation

In Sect. 3.1 we introduce the data assimilation problem as it appears in ocean forecasting. We describe ensemble-based data assimilation at the single level in Sect. 3.2 and continue in Sect. 3.3 to outline the suggested MLEnKF in a form that is applicable to models based on finite-volume schemes.

3.1 The data assimilation problem

Data assimilation is a statistical methodology that combines numerical models and observational data. To set up the problem, we start by considering a state variable $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$, which evolves forward in time by

$$\boldsymbol{x}(t + \Delta t) = \mathcal{M}(\boldsymbol{x}(t)) + \delta\boldsymbol{x}, \tag{18}$$

where $\mathcal{M}$ is now the model operator and $\delta\boldsymbol{x}$ is a random model error that can represent missing or unresolved physics in the model. In sequential data assimilation, the state is evolved by finite model time steps and we are interested in the assimilation of data at analysis time steps. Those are isolated model time steps where all quantities are at the same point in time. Hence, unless otherwise stated, we omit notation of time in the state vector for better readability, and we set $\boldsymbol{x} = \boldsymbol{x}(t)$.

We consider partial and noisy observations $\boldsymbol{y} \in \mathbb{R}^{n_y}$ of the true state, denoted $\boldsymbol{x}_{\mathrm{true}}$, of the simulated system, where the relationship between $\boldsymbol{y}$ and $\boldsymbol{x}_{\mathrm{true}}$ is given by

$$\boldsymbol{y} = \mathcal{H}(\boldsymbol{x}_{\mathrm{true}}) + \boldsymbol{\varepsilon}. \tag{19}$$

Here, $\mathcal{H}$ is the observation operator and $\boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{R})$ is the observation noise, which we assume to be Gaussian with a diagonal covariance matrix $\mathbf{R}$.

With the forward model and the observations in place, the data assimilation problem consists of updating the prior (forecast) probability density function $p(\boldsymbol{x})$ of the state $\boldsymbol{x}$ with the observations $\boldsymbol{y}$ to obtain the conditional (filtering) posterior probability density function $p(\boldsymbol{x}|\boldsymbol{y})$. This is done using Bayes' theorem, and in our situation the distributions are represented by ensembles. In the rest of this paper, we consider sequential data assimilation in terms of filtering, meaning that the model in Eq. (18) is only used to evolve the state between observation times.

3.2 Single-level ensemble Kalman filter

The EnKF is a popular technique for assimilating observation data into an ensemble. In this approach, covariances between the state and data variables are computed from the ensembles, and in doing so, one also forms a Monte Carlo approximation of the Kalman gain $\mathbf{K}$. The state vectors in the ensemble are then linearly updated, see, e.g., Evensen (2006) and Carrassi et al. (2018). In sequential data assimilation routines, this procedure is then repeated at all data gathering times.

The mean and the perturbations in the observation space without noise are

$$\overline{\boldsymbol{y}} = \frac{1}{N} \sum_{e=1}^{N} \mathcal{H}(\boldsymbol{x}_e) \tag{20a}$$

and

$$\tilde{\boldsymbol{y}}_e = \mathcal{H}(\boldsymbol{x}_e) - \overline{\boldsymbol{y}}, \tag{20b}$$

respectively.

For the single-level EnKF, the Kalman gain can be approximated by

$$\mathbf{K}^{\mathrm{SL}} = \mathbf{\Sigma}_{XY}^{\mathrm{SL}} (\mathbf{\Sigma}_{YY}^{\mathrm{SL}})^{-1}, \tag{21}$$

with

$$\mathbf{\Sigma}_{XY}^{\mathrm{SL}} = \frac{1}{N-1} \sum_{e=1}^{N} \tilde{\boldsymbol{x}}_e (\tilde{\boldsymbol{y}}_e)^\top \tag{22a}$$

and

$$\mathbf{\Sigma}_{YY}^{\mathrm{SL}} = \frac{1}{N-1} \sum_{e=1}^{N} \tilde{\boldsymbol{y}}_e (\tilde{\boldsymbol{y}}_e)^\top + \mathbf{R}. \tag{22b}$$

Here, $\mathbf{\Sigma}_{XY}^{\mathrm{SL}}$ estimates the cross-covariance between the state and the observations, while $\mathbf{\Sigma}_{YY}^{\mathrm{SL}}$ is the covariance between the observations. Note that the construction of a large state covariance matrix is circumvented in Eq. (22), and that for linear observation operators $\mathbf{H} \in \mathbb{R}^{n_y \times n_x}$, we can write $\mathbf{\Sigma}_{YY}^{\mathrm{SL}} = \mathbf{H}\mathbf{\Sigma}_{XY}^{\mathrm{SL}} + \mathbf{R}$. The ensemble is then updated by

$$\boldsymbol{x}_e^a = \boldsymbol{x}_e + \mathbf{K}^{\mathrm{SL}} (\boldsymbol{y} + \boldsymbol{\varepsilon}_e - \mathcal{H}(\boldsymbol{x}_e)), \quad e = 1, \dots, N, \tag{23}$$

where the superscript a marks the posterior (often called the analysis) states, and $\boldsymbol{\varepsilon}_e$ are independent realisations of the observation noise. Various algebraically equivalent forms of the Kalman update exist and in practice the cross-covariance or the Kalman gain matrix is often tapered away from observation indices to avoid spurious correlations that are theoretically irrelevant but naturally appear because of sampling errors.

In the reminder of this paper, we assume sparse observational data, meaning that the $n_y \ll n_x$. As a result, the EnKF update formulated by Eq. (23) using the Kalman gain given by Eq. (21) is computationally feasible and does not represent a bottleneck in our data assimilation system. If, conversely, the dimensionality of the observations was large, the formulation in Eq. (23) would require the computation and storage of prohibitively large covariance matrices. To avoid this problem, various other formulations of the EnKF exist, such as the (local) ensemble transform Kalman filter (Bishop et al., 2001; Hunt et al., 2007), but these methods are outside of the scope of this paper and we instead choose to focus on the multi-level aspects for the situation with sparse data.

3.3 Multi-level ensemble Kalman filter

In the single-level EnKF, all ensemble members are defined on the same computational grid and the forward step is performed on the original grid for the entire ensemble. However, in the multi-level case, a multi-level ensemble is defined as in Sect. 2 to exploit coarser, less accurate, grid resolutions with smaller simulation cost in the forward step. Similar to the model operator, the model errors $\delta\boldsymbol{x}^l$ are projections from the original noise on the finest resolution. Therefore, MLDA provides a framework that allows the assimilation of data into a forecast from a multi-level ensemble.

The application of the multi-level estimator for the Kalman gain approximation within spatio-temporal problems was originally suggested in Chernov et al. (2021). Our work builds on this, and we revise the method for hierarchically gridded data to clarify the construction of the matrix operators. Note that the Kalman gain $\mathbf{K}^{\mathrm{ML}} \in \mathbb{R}^{n_x \times n_y}$ is a matrix at the finest resolution. At the single level, it is constructed by the multiplication of the state-observation cross-covariance and the inverse of the observation covariance from Eq. (21). We now draw the attention to these matrices in the multi-level situation. Using the multi-level estimator for $\mathbf{\Sigma}_{XY}$, we arrive at

$$\mathbf{\Sigma}_{XY}^{\mathrm{ML}} = \frac{1}{N^0 - 1} \sum_{e=1}^{N^0} \tilde{\boldsymbol{x}}_e^0 (\tilde{\boldsymbol{y}}_e^0)^\top + \sum_{l=1}^{L} \frac{1}{N^l - 1} \sum_{e=1}^{N^l} \left(\tilde{\boldsymbol{x}}_e^{l+} (\tilde{\boldsymbol{y}}_e^{l+})^\top - \tilde{\boldsymbol{x}}_e^{l-} (\tilde{\boldsymbol{y}}_e^{l-})^\top \right), \tag{24}$$

where differences between partner levels are taken on the finer resolution and contributions of the different levels are upscaled in the end. Similarly, the observation covariance matrix can be calculated from Eq. (11)

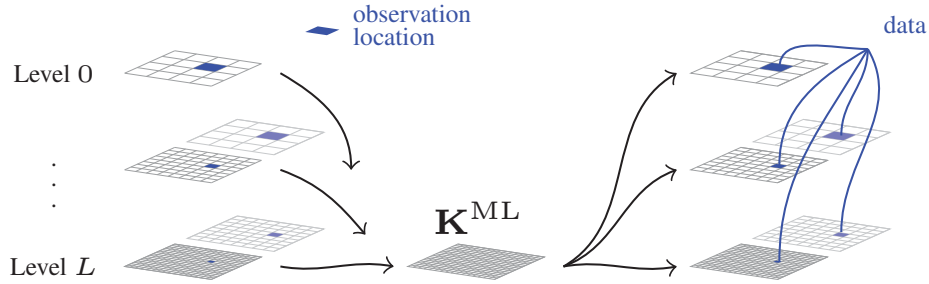


Figure 1: In multi-level conditioning with the MLEnKF, the empirical Kalman gain is assembled by a telescopic sum from the multi-level prior ensemble. Here, we use single observation location for illustration purposes. The Kalman gain is represented on the finest resolution and then projected to each level for updating. The update combines the prior ensemble with the data weighted by the Kalman gain for each ensemble member.

or, in the case of linear observation operators, extracted from $\boldsymbol{\Sigma}_{XY}^{\mathrm{ML}}$. Then, the multi-level Kalman gain becomes

$$\mathbf{K}^{\mathrm{ML}} = \boldsymbol{\Sigma}_{XY}^{\mathrm{ML}}(\boldsymbol{\Sigma}_{YY}^{\mathrm{ML}})^{-1}. \tag{25}$$

For updating the multi-level ensemble, the Kalman gain is downscaled onto the respective level. For the coarsest level of the MLEnKF, the analysis state then becomes

$$\boldsymbol{x}_e^{0,a} = \boldsymbol{x}_e^0 + \mathbf{K}^{\mathrm{ML}}(\boldsymbol{y} + \boldsymbol{\varepsilon}_e^0 - \mathcal{H}(\boldsymbol{x}_e^0)), \quad e = 1, \dots, N^0 \tag{26a}$$

and for the pairs on the higher level it is

$$\boldsymbol{x}_e^{l+,a} = \boldsymbol{x}_e^{l+} + \mathbf{K}^{\mathrm{ML}}(\boldsymbol{y} + \boldsymbol{\varepsilon}_e^l - \mathcal{H}(\boldsymbol{x}_e^{l+})), \qquad \boldsymbol{x}_e^{l-,a} = \boldsymbol{x}_e^{l-} + \mathbf{K}^{\mathrm{ML}}(\boldsymbol{y} + \boldsymbol{\varepsilon}_e^l - \mathcal{H}(\boldsymbol{x}_e^{l-})), \quad e = 1, \dots, N^l. \tag{26b}$$

Here, the observation noise $\boldsymbol{\varepsilon}_e^l$ are independent realisations for all ensemble members on all levels, where pairs on higher levels again share the same realisation.

In the estimation, we upscale recursively to the next level. This means that for the computation of the difference between two levels $\boldsymbol{x}^{l+} - \boldsymbol{x}^{l-}$, we upscale $\boldsymbol{x}^{l-}$ to have the same resolution as $\boldsymbol{x}^{l+}$. Eventually, we end up with an estimator on the finest level. In the ensemble update step, the spatial dimension of the Kalman gain is then downscaled to the respective level, the observation dimension stays unmodified all the time. Moreover, simple algebraic transformations ensure that the order of scaling and evaluation of statistical quantities does not matter. For example, the covariance matrix can be formed on a coarse grid and then upscaled to a finer grid by applying the projection operator from one side and its transposed from the other. Alternatively, the states can first be upscaled to the finer grid before the covariance matrix is assembled. Both of these procedures give the same result.

Figure 1 shows a schematic illustration of conditioning with multi-level ensembles. Simulations of state variables and synthetic data (left) are generated on the coarsest level and at partner levels. States and synthetic data from all levels are used to specify the Kalman gain matrix on the original resolution level (middle). The Kalman gain enables linear updating of the ensembles on all resolution levels with respect to the data (right).

With this implementation of the MLEnKF, we stay close to the formulation of Chernov et al. (2021). But even though we employ an analogous estimation and updating scheme as Chernov et al. (2021), their theoretical assumptions no longer hold in our set-up, and overall it is unlikely for realistic applications to

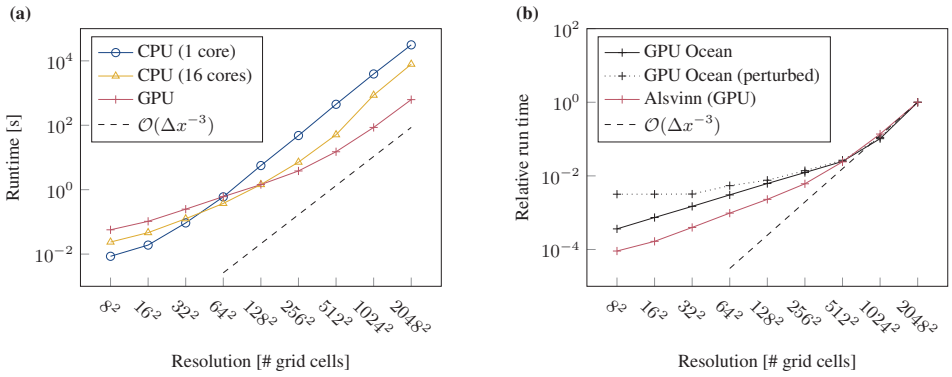


Figure 2: Practical computational work for different simulators with respect to number of grid cells. (a) Wall run time for Alsvinn on different computing infrastructures. (b) Computational performance for deterministic forward models using Alsvinn and GPU Ocean, and forward model with model error using GPU Ocean. Since Alsvinn and GPU Ocean solve similar but not identical problems, run times are given relative to the largest problem instance. Both figures include theoretical scaling as a dashed line for reference.

have a mean-field reference. Nonetheless, we aim for a computational speed-up through the MLEnKF, using more practical ways to assess the MLDA quality. The estimation over nested grids also shares similarities with Fossum et al. (2020), but in contrast to what is done in their paper, we avoid the hybridisation with Bayesian averaging and we keep the higher levels coupled.

4 Practical considerations

Until now we have discussed MLDA and its potential using MLEnKF on a theoretical level. In the following, we discuss some of the challenges and choices we face when applying MLEnKF to practical applications. This includes the configuration of the multi-level ensemble (Sect. 4.1), applying localisation across the different levels (Sect. 4.2), and dealing with the known problem of potentially negative eigenvalues in the estimated covariance matrices (Sect. 4.3).

4.1 Multi-level ensemble sizes

Multi-level data assimilation and forecast experiments rely on a fixed multi-level configuration throughout the experiments. In Sect. 2.2.3, we discussed how the MLMC configuration can be found by optimising the ensemble sizes for a given statistical accuracy τ^2. We will now discuss this aspect for practical data assimilation situations. For the multi-level setting, the ensemble sizes can be computed by Eq. (16), but they depend on i) the reduction in computational cost C^l towards lower levels l, and ii) the quantity of interest g that we want to estimate, along with its variance in (v^l) and between (V^l) the levels. We discuss each of these two dependencies separately.

4.1.1 Practical vs. theoretic computational work

We start by considering the scaling of the computational cost c^l for generating samples $\boldsymbol{x}^l$ on level l. More specifically, this means the computational performance of the forward model in Eq. (18), here represented

by the simulation framework used for solving the shallow-water equations. As discussed in Sect. 2.2.1, the theoretical scaling for the numerical model is $\mathcal{O}(\Delta x^{-3})$. This is however not always the case in practice. In particular, small problem instances on coarse grids can often exhibit suboptimal computational performance, either because time required by other elements in the simulation apart from the numerical scheme itself become a bottleneck, or because of hardware resources not being fully occupied. As a consequence, the optimal ensemble size obtained from Eq. (16) by using theoretic computational cost might not be optimal in practice.

To illustrate this, we consider two different simulation frameworks for solving PDEs using finite-volume methods with explicit time stepping. As such numerical schemes have an embarrassingly parallelisable structure, where each element of the output vector can be computed by the same numerical stencil and independently from the other elements, they can be implemented efficiently using massively parallel acceleration hardware such as graphics processing units (GPUs, see, e.g. Hagen et al. (2007)). The first one we consider here is the GPU-accelerated simulation framework GPU Ocean (Brodtkorb and Holm, 2021) designed for efficient simulation of single-level ensembles of simplified ocean models, which we will use in the later examples. The second one is Alsvinn (Lye, 2019), which is a more generic finite-volume framework specifically designed for MLMC applications, supporting execution on single- and multi-core CPUs as well as on GPUs.

Figure 2 shows the computational performance of the two simulation frameworks with respect to varying grid resolutions. In Fig. 2(a), we measure wall run times for Alsvinn for single-core CPU, multi-threaded CPU, and GPU. For large sized problems, we see that all these hardware configurations give perfect weak scaling (same slope as the dashed line) and that the GPU code is roughly ten and fifteen times faster than multi-core and single-core CPU, respectively. However, the performance on the GPU drops more rapidly than on the CPU when the problem size decreases. This is because the large number of computational cores on the GPU requires large problem instances to be fully utilised, and smaller problems lead to idle resources. The same pattern is also seen on a multi-core CPU, but at a later stage. As a consequence, even though the GPU gives best performance on large problems, the single-core CPU performs best on extremely coarse grids. For optimal MLMC speed-up, Alsvinn takes advantage of this by always using the best hardware for a given grid resolution, and thereby getting better scaling of computational cost than if only using the GPU. As a side note, we observe that the run time of a high-resolution simulation on the GPU is about the same as the run time of a CPU simulation on the next coarser level, meaning that a simulation pair is well-suited for parallel execution on different hardware.

In Fig. 2(b), we measure relative performance of the forward step from Eq. (18) with respect to the grid resolution using GPU Ocean both with and without additive model errors, keeping Alsvinn's GPU performance on a similar but not identical problem as a reference. Since GPU Ocean was developed for efficient high-resolution simulation of single-level ensembles, it only supports execution on GPUs and has perfect scaling for sufficiently large problems ($> 512^2$ grid cells). For coarser resolutions than 512^2, the performance drops for both frameworks to about $\mathcal{O}(\Delta x^{-1})$, which is expected when the problem size can be executed fully concurrently on the GPU and only the time stepping becomes dominant. When using additive model errors every 60 s (see details later in Sect. 5.1.2), we see that the performance drops even more for coarse grid resolutions. Although the model perturbations are negligible at high grid resolution, they become more and more dominating as coarser grids allow for larger finite-volume time steps. Eventually, the simulator alternates between one numerical time step and one model error perturbation, and the computational cost does not reduce any further.

From these results, we see how existing simulation frameworks do not offer theoretical performance scaling in practice, especially so when they utilise hardware accelerators such as GPUs. By only considering the single-core CPU implementation of Alsvinn, which has perfect scaling for a larger range of grid resolutions, it is possible to harvest the full potential speed-up of MLMC methods, but that speed-up would only be relative to high single-level computational costs. It is therefore beneficial to utilise the most efficient hardware option despite worse scaling.

In the following, we will consider both practical and theoretical values for computational cost. The practical cost will be used to determine the multi-level configuration that gives best speed-up using the

simplified ocean models with the forward models implemented in GPU Ocean today. The theoretical cost will show the upper bound for computational speed-up.

4.1.2 Variance analysis

We continue by considering the variances in Eq. (14), and we study how they are affected by the choice of quantity of interest. It is not straightforward to choose this quantity in the sequential data assimilation set-up, as estimates of various quantities of interest could be desired at different spatial locations and steps in time. For example, good estimation of multi-level ensemble mean is important to make a forecast. In contrast, the update steps from Eqs. (23) or (26) rely on the estimation of the Kalman gain which has to be constructed repeatedly from the covariance estimation. Also, we are interested in assessing the quality of the ensemble-based representation by evaluating various scores, e.g. the probability integral transforms.

With this in mind, we discuss two alternatives for the function g. In the final forecasts, the mean is often the primary quantity of interest, and hence one potential choice is

$$g_1(\boldsymbol{x}) = \boldsymbol{x}. \tag{27}$$

Alternatively, as the Kalman gain in Eq. (25) depends on the estimation of the covariance matrix which has the form $\mathbb{C}\mathrm{ov}[\boldsymbol{x}] = \mathbb{E}[g(\boldsymbol{x})]$ with

$$g_2(\boldsymbol{x}) = \tilde{\boldsymbol{x}}^2, \tag{28}$$

we use this definition of g in the analysis, where the vector entries are squared element-wise.

As the variances in Eq. (14) are a priori unknown, the values v^l and V^l must be estimated from a trial experiment for a given function g. In MLMC, the trial experiments simulate a number of samples per level to estimate these values, and we therefore construct a trial experiment for MLDA inspired by the corresponding MLMC set-up. Note that these experiments generate samples for estimating the variances, and therefore, they influence the configuration which must be chosen adequately to the data assimilation application. We start by defining the same number of ensemble members on every level, where one member per level shares the noise realisation with one member on all the other levels. We then run a trial data assimilation experiment, where we calculate the Kalman update on the finest level and project it to the coarser levels. Finally, we approximate the variances in Eq. (14) for each physical variable. Hereby, it is expected that the correlation between the levels decreases over the simulation time, and in a conservative manner, we hence evaluate the variances at the last assimilation time.

It would not be valid practice to run a trial experiment to determine the correlations between the levels for every new set of observational data before running the actual multi-level data assimilation. For this reason, we have repeated experiments using data from different stochastic realisations of the truth. We experience qualitatively similar results. We therefore believe that one can extrapolate the obtained results from one trial experiment as long as the trial experiment is sufficiently representative for the relevant application.

4.2 Localisation

Due to the finite sample approximation of the Kalman gain, spurious long-distance correlations are expected to degrade the quality of the updates. Localisation is a common strategy to improve the practical quality of EnKFs (Sakov and Bertino, 2010). Within MLEnKFs, localisation has also been studied for level-local formulations, where the levels do not share the same Kalman gain (Hoel et al., 2020). However, we will incorporate spatial localisation into the MLEnKF and the term "localisation" will in the rest of this article refer to spatial localisation. In the construction of the state covariance matrix, the entries corresponding to long-distance correlations are diminished and for the multi-level covariance estimator, we can pick this tapering for each level individually (Destouches et al., 2023). In particular, we employ Kalman-gain localisation (Chen and Oliver, 2017).

Let $\mathbf{w} \in \mathbb{R}^{n_x \times n_y}$ be a tapering matrix. In the updates in Eq. (23) or Eq. (26), we use the Schur product of the kernel and the Kalman gain $\mathbf{w} \circ \mathbf{K}$. Generally, the need for localisation primarily arises due to a small number of ensemble members. As the involved number of ensemble members varies per level in the

multi-level case, the MLEnKF offers the possibility to include localisation on each level independently. We incorporate this directly into the construction of the Kalman gain by

$$\mathbf{\Sigma}_{XY}^{\mathrm{ML,loc}} = \frac{\mathbf{w}^0}{N^0} \circ \sum_{e=1}^{N^0} \tilde{\boldsymbol{x}}_e^0 (\tilde{\boldsymbol{y}}_e^0)^\top + \sum_{l=1}^{L} \frac{\mathbf{w}^l}{N^l} \circ \sum_{e=1}^{N^l} \left(\tilde{\boldsymbol{x}}_e^{l+} (\tilde{\boldsymbol{y}}_e^{l+})^\top - \tilde{\boldsymbol{x}}_e^{l-} (\tilde{\boldsymbol{y}}_e^{l-})^\top \right), \tag{29}$$

where $\mathbf{w}^l$ entries are either downscaled values from $\mathbf{w}$ or 1. In this case, $\mathbf{\Sigma}_{YY}^{\mathrm{ML}}$ cannot be extracted from $\mathbf{\Sigma}_{XY}^{\mathrm{ML,loc}}$ as in Eq. (22b), but has to be calculated on its own by Eq. (11b). The level-wise application for localisation leads to a biased estimator in theory, but localisation has shown to be a powerful heuristic in practice (Carrassi et al., 2018).

4.3 Negative eigenvalues

It is well-known for MLMC that the covariance estimate in Eq. (11b) can attain negative eigenvalues, see, e.g. Shivanand (2023) and the references therein. This arises from the fact that the empirical measure associated with an MLMC estimator is not positive in the sense that all Dirac contributions have a positive sign. In the scalar-case, this means, for example, that approximated variances can become negative, which is statistically invalid and would break down the mathematical framework. As the measure associated with the single-level estimator is positive, this has not been a prevalent issue in single-level data assimilation.

Methods to construct positive semi-definite matrices are expensive (Maurais et al., 2023) and the calculation of eigenvalues for large matrices can be prohibitively costly. In our setting with sparse observations, we never assemble the full covariance matrix $\mathbf{\Sigma}_{XX}^{\mathrm{ML}}$. Instead, we need to factorise the low-dimensional matrix $\mathbf{\Sigma}_{YY}^{\mathrm{ML}}$ in the suggested MLEnKF approach. With low-dimensional observation data, its eigenvalues are fast to compute.

Not surprisingly, we noticed rare occasions of slightly negative eigenvalues in our experiments, and we identify these to be associated with situations where the observation locations lie on a wave front. If no mitigation strategy is implemented and $\mathbf{K}^{\mathrm{ML}}$ is used directly in the update, the ensemble shows non-physical behaviour over time and eventually becomes useless. Visually, strong waves in opposite directions take over to dominate the dynamics. For plain MLMC this is less of a problem, but it becomes pivotal for data assimilation within the MLEnKF and requires special attention as corrupted updates affect the future ensemble forecasts. Since this seems to happen only rarely in our experiments, we overcome this issue simply by skipping the data assimilation to avoid non-physical properties in the multi-level estimation, when the smallest eigenvalue $\mathbf{\Sigma}_{YY}^{\mathrm{ML}}$ becomes negative. By doing so, we have noticed that the ensemble results stayed well calibrated and physically reasonable. We monitored slightly negative eigenvalues in less than 1:10.000 times of the assimilated observations, such that missed assimilation becomes negligible.

Alternatively, one could replace the Kalman gain by a single-level Monte Carlo estimator from one of the levels in case of negative eigenvalues, but we have seen that it is most important to keep updates statistically and physically sound. In mitigation strategies, it is not enough to manipulate only $\mathbf{\Sigma}_{YY}^{\mathrm{ML}}$, but $\mathbf{\Sigma}_{XY}^{\mathrm{ML}}$ should also be corrected.

5 Application to simplified ocean model

We now study the results from assimilating sparse observations into a simplified ocean model using MLEnKF. We consider an identical-twin experiment similar to that of Holm et al. (2020b) and Beiser et al. (2024). This synthetic case is described in Sect. 5.1. Here, both the nature run (sometimes also referred to as the ground truth) that generates the data and the single- and multi-level ensembles will be initialised by an unstable steady-state solution, which become turbulent when adding small random perturbations. One purpose of these experiments is to assess the potential speed-up of the multi-level framework, and in Sect. 5.2 we find the optimal configuration of the multi-level ensemble given a statistical accuracy obtained from a single-level experiment. We then investigate the properties of MLDA experiments by comparing them to single-level experiments (Sect. 5.3).

5.1 Problem description

We outline the design of our numerical experiments by first describing the mathematical model, initial conditions, model perturbations, and grid resolutions. Then, we turn to the high-level structure of the data assimilation experiment and describe the available observations.

5.1.1 Shallow-water model and initial conditions

The shallow-water model is a non-linear hyperbolic system of equations for conservation of mass and momentum in two dimensions given by

$$\begin{bmatrix} \eta \\ hu \\ hv \end{bmatrix}_t + \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}_x + \begin{bmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}_y = \begin{bmatrix} 0 \\ fhv \\ -fhu \end{bmatrix}, \tag{30}$$

where the subscripts denote partial derivatives. Here, η represents the sea surface elevation as a deviation of the equilibrium mean, and hu and hv are the moments in x- and y-directions, respectively. The equilibrium depth is given by H, so that the total water depth becomes $h = H + \eta$. Furthermore, g and f are the gravitational constant and Coriolis parameter, respectively. To represent the forward model $\mathcal{M}$, we solve Eq. (30) using a high-resolution central-upwind finite-volume scheme based on Chertock et al. (2017), as implemented in the GPU-accelerated simulation framework of GPU Ocean (Brodtkorb and Holm, 2021).

The numerical experiments and initial state are inspired by a test case suggested by Galewsky et al. (2004) for validating shallow-water models on a rotating sphere. We consider a rectangular domain of size 1332 km $\times$ 666 km and periodic boundary conditions, initialised with a westward and eastward jet in the northern and southern part of the domain, respectively. Both jets are initially in a steady-state with respect to geostrophic balance, meaning that they satisfy the relation

$$hu = -\frac{gh}{f}\frac{\partial \eta}{\partial x} \quad \text{and} \quad hv = \frac{gh}{f}\frac{\partial \eta}{\partial y}. \tag{31}$$

The system is however unstable, meaning that even a small perturbation will lead to developments of turbulent and complex currents and eddies over time. By introducing perturbations through random sampling of the model error term, the currents of different model realisations will evolve in different ways. Furthermore, we use parameters inspired by the conditions in the Barents sea, using $H = 228$ m as equilibrium depth, along with $g = 9.806\ \frac{\text{m}}{\text{s}^2}$ and $f = 1.405 \cdot 10^{-4}\ \frac{1}{\text{s}}$. For more details, see Appendix A.

5.1.2 Sampling of model errors

The random model errors $\delta \boldsymbol{x}$ collects perturbations $\delta\eta, \delta hu, \delta hv$ for each of the physical variables. We generate the noise by first sampling $\delta\eta$ from a Karhunen-Loeve-type random field, and then computing δhu and δhv according to the geostrophic balance in Eq. (31) to ensure that $\delta \boldsymbol{x}$ does not just contain gravitational waves. With a scaling parameter Θ and a decay parameter θ, the perturbation in η is created by

$$\delta\eta = \Theta \sum_{i=i_0}^{i_0+I} \sum_{j=j_0}^{j_0+J} \alpha_{i,j}^{\sin}\ i^{-\theta} j^{-\theta} \sin(2i\pi\zeta)\sin(2j\pi\xi) + \alpha_{i,j}^{\cos}\ i^{-\theta} j^{-\theta} \cos(2i\pi\zeta)\cos(2j\pi\xi), \tag{32}$$

where $\alpha_{i,j}^{\sin}, \alpha_{i,j}^{\cos} \in \mathcal{U}[-1,1]$. We here parameterise the domain as $\zeta, \xi \in [\alpha^x, 1+\alpha^x] \times [\alpha^y, 1+\alpha^y]$ and introduce the shifting $\alpha^x, \alpha^y \sim \mathcal{U}[0,1]$ to break up spatial patterns. Note that $\delta\eta$ is continuous and it is evaluated in the cell centres to get discrete realisations. For the rectangular domain, we set $i_0 = 1$ and $j_0 = 2$ to obtain basis functions with the same physical scales in x and y directions. Moreover, we use $I = J = 7$ basis functions per axis, $\theta = 0.9$ for the decay, and $\Theta = 0.001$ for the scaling. After sampling $\delta\eta$ on the given grid, we compute discrete values for δhu and δhv from Eq. (31) using central differences for the derivatives.

For multi-level methods, it is essential that the partners in a pair share the same realisation. When the stochasticity enters the system only via initial conditions or external forcing, this coupling is rather easy to

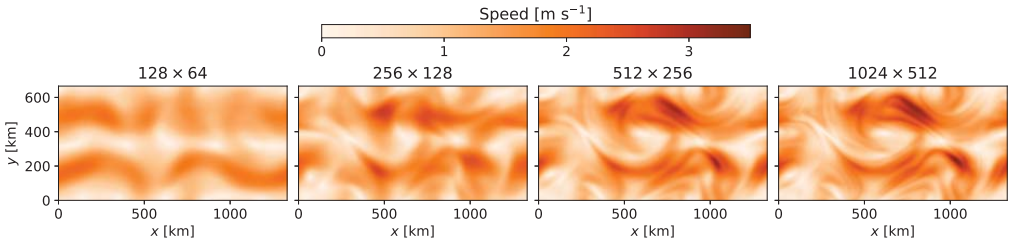


Figure 3: Same realisation of the shallow-water model for different resolutions after simulation until $t = 10$ d.

implement. In our case, however, we have frequent perturbations working directly on the state variables, which means that the two partner simulations should share the same realisation of the model error on their different levels. It is thus necessary to evolve coupled simulators to the same time and transfer information about the sampling. Since the performance of the simulation of a coupled pair should not be affected much by the model error, we take care that only some data for the sampling is transferred between the partners.

The necessary information to sample a single realisation of $\delta\boldsymbol{x}$ is contained in the $2IJ$ random numbers $\alpha^{\sin}_{i,j}, \alpha^{\cos}_{i,j}$ and α^x and α^y. Therefore, the same realisation of the model error can be easily applied to coupled simulations on different levels by communicating only these numbers and evaluating Eq. (32) in the cell centres of the corresponding grid.

For our numerical experiments, we employ a model time step $\Delta t = 60$ s for Eq. (18) independent of the grid resolution of the model. This means that states on all levels are perturbed according to the model error after every minute, and that the numerical model $\mathcal{M}$ is takes as many iterations of the numerical schemes as required to achieve stability.

5.1.3 Grid resolutions for the forward model

We now define the various grid resolutions that are used to represent the levels in our multi-level experiment. The finest grid resolution representing level L in all our experiments, is a grid consisting of 512×1024 cells, which corresponds to a horizontal resolution of 1300 m. We then define the other levels based on a factor two coarsening from level L, meaning that they will consist of 512×256, 256×128, and 128×64 grid cells. This means that the horizontal resolution of the coarsest grid becomes 10.4 km.

To first get an impression of how the relevant dynamics are maintained by different grid resolutions in our case, we run a single realisation of the model on each of the four levels using the same realisation of the model error. Figure 3 shows the resulting currents at $t = 10$ d. Here, the currents on the two finest levels, 1024×512 and 512×256, are almost the same with only minor details on the sharpness of some features. With 256×128 grid cells, the main structures are still intact, although most of the details are smoothed out. On the coarsest resolution 128×64, however, we no longer recognise the shape of the currents from the finest resolution, and the solution contains few details in general.

5.1.4 Data assimilation experiment and sparse observations

We consider identical-twin experiments, where ensemble members and the nature run generating the data are all sampled using the same mathematical and numerical model, initial conditions, and distribution for sampling model errors. The experiments consist of three distinct parts. First, there is a spin-up of three days, allowing the turbulent dynamics to start evolving and give an initial spread to the ensemble. Then, we have seven days of data assimilation. Finally, we generate a forecast for three days.

The nature run state $\boldsymbol{x}_{\text{true}}$, is generated on the finest grid resolution, consisting of 512×1024 cells. Figure 4 shows the magnitude of the particle velocities at the initial conditions, and after three, six, and ten days, illustrating how the jets develop instabilities.

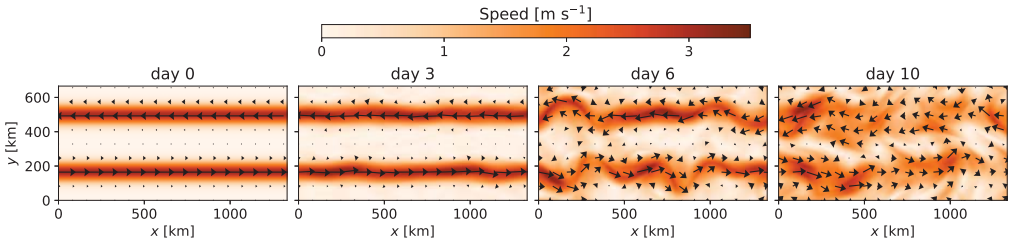


Figure 4: The synthetic truth at different times during the experiment: Starting with a westward jet in the north and an eastwards in the south, turbulent dynamics builds up over the following days and becomes clearly visible from day 6.

From $\boldsymbol{x}_{\text{true}}$, we get observations of (hu, hv) at 50 regularly distributed locations throughout the domain at every 15 min between day 3 and day 10. This means that the shortest distance between any two neighbouring observation locations is about 100 km. Using Eq. (19), the observation $\boldsymbol{y}$ consists of $N_y = 100$ elements extracted directly from the state space by a linear observation operator $\mathbf{H}$, leaving η entirely unobserved. Furthermore, we use observation error $\epsilon \sim N(0, \mathbf{R})$ with $\mathbf{R} = 500\mathbf{I}_{N_y}$, where $\mathbf{I}_{N_y}$ is the $N_y \times N_y$ identity matrix. With the given equilibrium depth of the model, this value for $\mathbf{R}$ corresponds to a standard deviation of approximately 0.1 $\frac{\text{m}}{\text{s}}$ for u and v.

In the data assimilation steps, of both single-level and multi-level approaches, we handle observations serially using localisation. For this we specify a localisation radius of 50 km. We further apply relaxation-to-prior with a factor of 0.25 (Zhang et al., 2004). For details on these choices, see Beiser et al. (2024).

5.2 Multi-level configuration

We now aim at designing and configuring multi-level ensembles that can offer a computational speed-up over single-level ensembles for this data assimilation experiment. Recall from Sect. 4.1, that to find the optimal configuration of the multi-level ensemble sizes, we need to experimentally assess the computational cost of running the model at each level c^l, the variance at each level v^l for a given quantity of interest g, and the corresponding variance between partner levels V^l.

We start with determining the computational costs per level. For the GPU Ocean code, we saw in Sect. 4.1.1 that we only maintain theoretic computational performance for problem instances with more than 512^2 grid cells. We therefore measure the practical cost of running the model for the grid sizes that we will use in our multi-level experiments to determine the computational work C_l. These values behave qualitatively very similar to the curve in Fig. 2 where the considered resolutions are in the range close to the theoretical scaling between the two finest levels, but the curve flattens out for the coarser levels. Because the relative high computational cost for coarse simulations is somewhat special for GPU-accelerated code, and because we also want to discuss the practicability of multi-level data assimilation in general, we will consider both the theoretical cost and practical cost when configuring the multi-level ensemble size. The assumption of theoretic computational cost then represents the speed-up potential through using multi-level statistics if obtainable through other software frameworks.

We continue with the variance analysis as in Sect. 4.1.2, quantifying the effects that have been indicated in Fig. 3. Figure 5 shows the results of a trial experiment for two different quantities of interest. In panel (a), we use the mean, as given by g_1 from Eq. (27), whereas in (b), we consider the covariance, represented by g_2 from Eq. (28). Hereby, the variances are recorded per variable, such that the columns in the figure correspond to η, hu and hv. In all panels, the curve for v^l is increasing, indicating that many details are lost at the coarser levels. Also, all curves for the variances between the levels V^l lie below v^l, but they do not have a strongly decreasing slope. For the function g_2 in panel (b) both curves are even closer together than

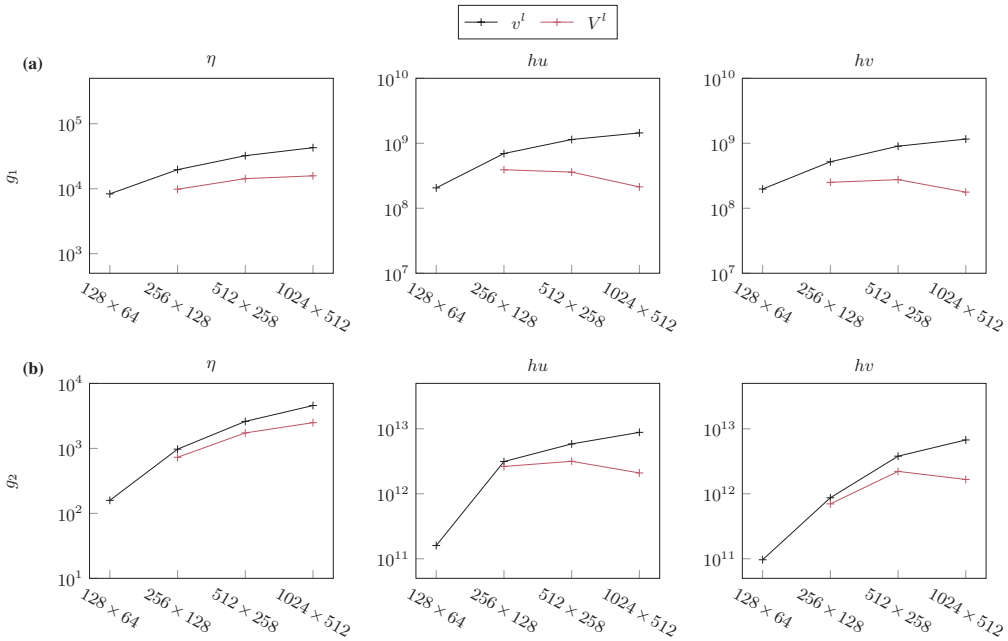


Figure 5: Variance analysis per variable as function of grid resolution. Dark lines represents the variance per level and light lines are variance between the partners after 10 days of trial experiment for (a) g_1 and (b) g_2. Here, g_1 represents the mean as quantity of interest and g_2 the variance.

Table 1: Ensembles with the same theoretical error as a single level ensemble with $N_e = 50$ in relation to the listed functional.

Case Name	Work for C^l	Function for V^l	SWE variable	**Ensemble Size** 1024×512	512×256	256×128	128×64	**Speed Up** Factor
(A)	theoretical	$g_1(\boldsymbol{x}) = \boldsymbol{x}$	hu	13	46	134	275	2.3
(B)	practical	$g_1(\boldsymbol{x}) = \boldsymbol{x}$	hu	14	86	-	-	1.7
(C)	theoretical	$g_2(\boldsymbol{x}) = \tilde{\boldsymbol{x}}^2$	hu	19	66	185	-	1.7

for g_1 in panel (a). We note that the relative changes between the levels are most relevant for the ensemble size, and not the actual magnitudes. In the data assimilation experiment, only data for the momentum was assimilated and the plots for hu and hv are qualitatively similar per function, but for η it looks different.

For a speed-up through multi-level estimation, it would be beneficial if the computational costs scaled with a steep slope and if the fine levels had very strong correlation (small values for V^l relative to v^l). While this experiment is not an ideal scenario for maximal speed-up through multi-level estimators, these kinds of variances are more likely to be faced in many practical scenarios. In this sense, the case is on the edge for the multi-level approach to be beneficial.

Based on the computational costs and the variance analysis, where the latter depends on the chosen quantity of interest, we set the multi-level ensemble sizes as in Sect. 2.2.3. Here, we relate to a single-level ensemble with 50 member and consider several multi-level configurations. For the computational costs, we consider both theoretical scaling and the practical performance and for the variances, we consider two different quantities of interest. The variances are evaluated for each primary variable, but we pick hu for further analysis, since the dynamics in the model is dominated by motion along the x-axis. We estimate τ according to Eq. (5) for the relevant quantities of interest and then set multi-level ensemble sizes according to Eq. (16).

As a result, Table 1 lists three different multi-level configurations along with their corresponding speed-up. To have a representative set of sufficiently different experiments, we chose different number of levels for each of the cases. For these multi-level ensembles, the computational work can be calculated by Eq. (17), while the corresponding computational cost for the single-level ensemble with 50 members is $C^{\mathrm{SL}} = 50\ c^L$ according to Eq. (4). The speed-up in the forward step is then the factor

$$\frac{C^{\mathrm{SL}}}{C^{\mathrm{ML}}}, \tag{33}$$

as listed in the rightmost column in Table 1. The computational advantage of the multi-level ensemble in these experiments comes by design.

It is crucial to note that multi-level methods do not always yield a speed-up, as this depends on the characteristics of the problem represented by C^l and V^l. For the combination of practical work and function g_2, for example, we do not obtain a speed-up for any number of levels. We have therefore not included that combination for the subsequent numerical experiments.

5.3 Numerical results

Using the multi-level ensembles from Table 1, we now run the data assimilation experiment presented in Sect. 5.1. First, we aim to evaluate whether the results of MLDA are in accordance with single-level results. Then, we demonstrate how the multi-level hierarchy can interplay with localisation and investigate how data assimilation influences the multi-level eligibility. Last, we assess the MLEnKF by means of rank histograms.

In previous studies with analytical examples, the convergence of the MLEnKF against mean-field EnKF distributions was analysed, but such a complex example as in this article is to the best of the authors knowledge not yet covered by any theoretical results.

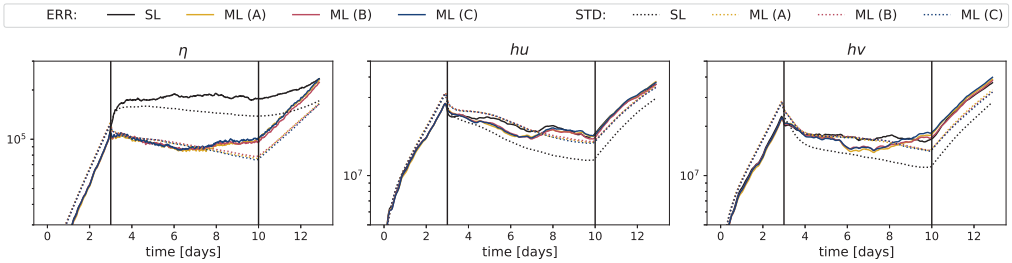


Figure 6: ERR and STD during the data assimilation experiment as in Sect. 5.1 for a single-level ensemble with 50 members (SL) and the multi-level cases (ML) from Table 1. The curves are smoothed with a moving average over 3 h to reduce the noise in the plot. Vertical lines mark transitions between spin-up, data assimilation, and forecast.

5.3.1 Error of single-level and multi-level ensembles

We start by considering prediction error versus the nature run for both single-level and multi-level ensembles. Throughout the data assimilation experiment, we assess the error in the mean (ERR) of the ensemble per variable

$$\text{ERR} = \|\boldsymbol{\mu}[\boldsymbol{x}] - \boldsymbol{x}_{\text{true}}\|_2, \tag{34}$$

where the mean estimate $\boldsymbol{\mu}[\boldsymbol{x}]$ comes from a single-level ensemble as in Eq. (2) or from a multi-level ensemble as in Eq. (11). Note that the criterion in Eq. (34) was not directly used in the design of the multi-level estimators, but it gives insight about the state estimation resulting from the data assimilation. In addition to ERR, we also consider the ensemble spread measured by the empirical standard deviation (STD).

In Fig. 6, we show the ERR (solid) and STD (dotted) over time for the four different experiments and for each of the physical variables. Keep in mind that the phase before day 3 is spin-up and that the phase after day 10 is plain forecast. The black lines represent a single-level experiment with $N_e = 50$, whereas the other three represent the three multi-level configurations from Table 1. For the single-level experiment, we see that the STD lies under the ERR, but it follows the same trends. For the multi-level experiments, they all show the same quality in terms of ERR for hu and hv (middle and right display), confirming that the assumption of same statistical errors in the ensembles was legitimate. For the unobserved variable η (left display), the error for the single-level ensemble on the finest level is quite noisy, but the multi-level ensemble smooths this out. Surprisingly, the ERR appears to increase at around day 7 for all multi-level configurations and the single-level experiment. In contrast, the STD declines all through the data assimilation period. Studying replicate runs, this increase in ERR does not seem to reflect only Monte Carlo error, but it is more likely an effect of increased turbulence in the forward model (see Fig. 4). Ensemble-based data assimilation methods are not exact, and with complex non-linear dependencies in the model, they can cause bias or underestimation of uncertainty. Here, after day 7, the data assimilation continues to give a steady decline in STD while it induces a bias in the state estimates. In our experiments, the application of MLEnKF instead of single-level EnKF does not affect the results for the currents. More surprisingly, interchanging g_1 and g_2 does not change the results for ERR and STD much either. All results for ERR of the moments are essentially equivalent, but the STD is a bit higher for the multi-level cases.

The purpose of the experiment is to investigate whether MLDA results agree with the equivalent single-level data assimilation result. It can be clearly seen that the MLDA errors concur with the errors of the single-level experiment. Due to high cost of a single experiment in this set-up, it is prohibitive to evaluate the statistical error by repetitions of the full experiment. Nonetheless, we have repeated the entire data assimilation experiment several times and recorded results with the qualitative conclusion that the multi-level and single-level experiment achieve equivalent outcomes.

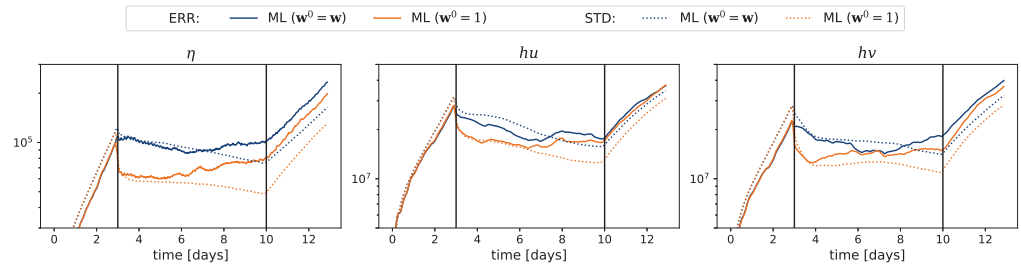


Figure 7: ERR and STD during the data assimilation experiment for the multi-level ensemble (A) with (blue) and without (orange) localisation on the coarsest level.

5.3.2 Localisation effects in the MLEnKF

The MLEnKF offers the possibility to integrate localisation per level, a concept that is not realisable with a single-level ensemble. While it is not practical to abstain from localisation with only 50 ensemble members in the single-level case, we demonstrate the effects of level-wise localisation in the MLEnKF by disregarding localisation for the coarsest level of case (A) with 275 members.

In Fig. 7, we show the comparison of localisation on all levels (blue) and localisation on all except for the coarsest level (yellow). There cannot be an equivalent single-level experiment for this approach. Comparing the ERR for the two experiments, we notice that the ensemble without localisation on the coarsest level is updated more strongly towards the truth in the beginning of the DA period. This is because the updates are not tapered but rather extend over the entire domain. The standard deviation is also reduced by a similar magnitude in this early phase. However, once the dynamics becomes more turbulent around day 6 to 7, the curve for ERR without localisation on the coarsest level approaches the other experiment. In this unstable regime, small spurious correlations seem to be enough to cause deviations from the truth. The ensemble without localisation at the coarsest level keeps the lower STD during the remaining course of the experiment.

5.3.3 Multi-level eligibility

The computational advantage of MLMC is instantiated when the variances of the details represented by V^l are much smaller than the general variance in the estimator, where v^l are references for the latter one. See Eq. (14) for the definitions and note that this has not been relevant for single-level estimation. The MLEnKF interferes with these variances and the theory does not guarantee that the updates will be favourable for multi-level speed-up. Therefore, we record the ratio V^l/v^l over the time span of the data assimilation experiment compared to the course of an MLMC experiment.

In Fig. 8, we plot the relative variance of the difference in the function g at level l compared with the variance of g at the same level, e.g. V^l/v^l. The dotted lines represent the relative variance in a MLMC experiment without any data assimilation, whereas the solid lines show the situation with data assimilation using the MLEnKF. In all three displays (η, hu, hv), the solid lines tend to go above the dotted line when data assimilation starts after 3 d of spin-up. Data assimilation reduces the uncertainty in the ensemble, and this means the values for v^l (denominator here) will go down. Going from dark blue (fine level) to light blue (coarse level), the variance in the difference gets relatively larger compared with the variance in the function, especially in the beginning of the data assimilation period. The exception is the coarsest level, where the solid and dotted lines are much more similar. This indicates that the sparse observations have mainly local effect, and the relative differences are hence largest at the finer levels. The increase of the MLDA curves stagnates over the data assimilation period.

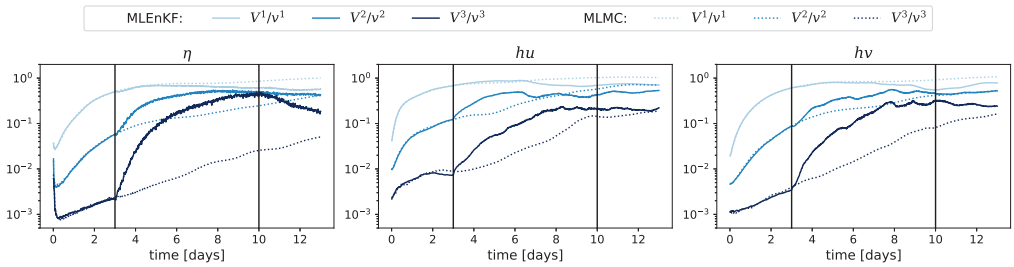


Figure 8: Relative variance of the level differences to the variance on those levels over the time of the data assimilation experiment (A). MLMC experiment with same ensemble size for reference.

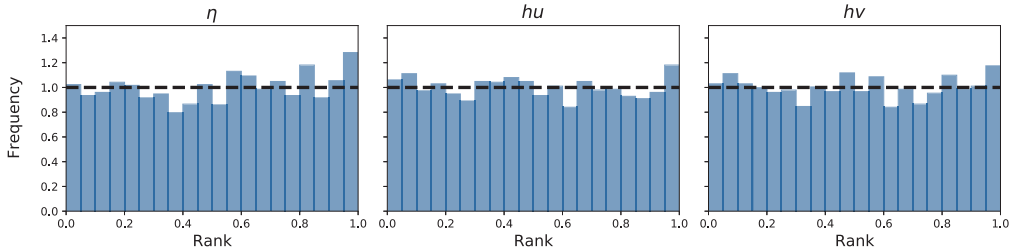


Figure 9: Rank histograms from multi-level ensemble forecasts for each primary variable generated by more than 3000 observations. The dotted line represents a uniform distribution for reference.

5.3.4 Rank histograms

Here, we aim to assess the distribution in the analysis ensemble produced by the MLEnKF. For this purpose, rank histograms are a common technique to visually evaluate sample and ensemble spread, where uniform histograms are usually considered desirable (Hamill, 2001). In a similar study, we have already shown that a single-level EnKF produces flat rank histograms for numerical experiments similar to the one used in the current paper (Beiser et al., 2024).

We choose a multi-level ensemble of sizes 12, 37 and 88 on the three highest levels, corresponding to a single-level ensemble with 40 members for g_1 and practical work. For each of these replicate experiments, we initialise a new nature run and a new ensemble using the same parameters as before. We then spin the ensemble up for three days, followed by one day with data assimilation and three hours of forecast. At the end of the forecast, we record the values of the nature run and the ensemble at 24 different locations in the domain. These locations are chosen by a regular equidistant 4×6 pattern, and are assumed to be uncorrelated within the time frame of the data assimilation and forecast.

For multi-level ensembles, the formulation by a probability integral transform value is useful (Gneiting et al., 2007), where a rank corresponds to the nature run's value in the empirical cumulative distribution function represented by the ensemble. Therefore, the multi-level ensemble is used to resample a corresponding single-level ensemble which is used for the evaluation of the probability integral transform value (Gregory and Cotter, 2017). The size of the resampled ensemble is independent and we choose 1,000 samples per multi-level ensemble.

Figure 9 visualises the histograms for 138 repeated rank experiments using bins of width 0.05. Altogether, this means $138 \cdot 24 \approx 3000$ testing observations. The histograms for hu and hv are close to a uniform distribution indicating that the statistics of the multi-level posterior ensemble are reliable in the sense that

nature run ranks are spread just like they would with the true (unavailable) posterior solution. While the histogram for η has some outliers in the upper end, the general rank histogram is still close to the theoretically correct uniform distribution, and we notice no indication of systematic errors in the data assimilation with the MLEnKF.

6 Multi-level drift trajectory forecasting

In the previous section, we explored the multi-level state estimation problem through data assimilation, and we assessed the quality of the approximate posterior distribution of the ocean states directly. We now discuss how we can use the multi-level ensemble to forecast derived quantities of interest, specifically through probabilistic forecasting of drift trajectories. The approach for constructing drift trajectories in the multi-level setting is outlined in Sect. 6.1, while the results are shown in Sect. 6.2.

6.1 Multi-level drift trajectory model

Drift trajectories are typically computed either online as part of the dynamical ocean model, or offline through dedicated drift trajectory simulation software that takes the output files from the dynamical ocean simulator as input (Dagestad et al., 2018; Delandmeter and van Sebille, 2019). Here, we consider the online approach using a passive drift model. We let $\Psi(t) \in \mathbb{R}^2$ contain the drifter's x and y coordinates at time t, and update its position by

$$\Psi(t + \Delta t) = \Psi(t) + \Delta t \begin{bmatrix} u(t, \Psi(t)) \\ v(t, \Psi(t)) \end{bmatrix}. \tag{35}$$

Here, the time step is the same as the time step for the simplified ocean model, and u and v are bi-linear reconstructions of the simulated ocean currents in space to best represent the exact location of the drifter at time t.

Using a single-level ensemble of dynamical models, the simplest way to produce a probabilistic trajectory forecast is to simulate one drifter per ocean ensemble member and solve Eq. (35) for each of them. In a multi-level ensemble, however, the individual ocean states are no longer realisations from the statistical distribution they collectively describe, and the "one drifter per ocean model" approach becomes much harder. While it is possible to generate a multi-level drifter ensemble from the already generated posterior multi-level ensemble generated in this paper, it is difficult to generate likelihood plots (equivalently, 2D histograms or 2D probability density functions). Techniques such as the maximal entropy method are transferable to multi-level ensembles, but they often suffer from Newton convergence problems for multi-level ensembles (Bierig and Chernov, 2016). We therefore concentrate on generating a single-level drifter ensemble from the already computed multi-level ensemble, and remark that the computational runtime of the drifters is insignificant compared to the total runtime of the ensemble of ocean states. We then construct an ensemble of N_d drifters that behave according to the statistical estimates of the currents obtained from the multi-level ensemble. One such option is to let each drifter realisation be advected according to a random walk by sampling

$$u_d(t, \Psi_d(t)) \sim \mathcal{N}\Big(\mu_u\big(t, \Psi_d(t)\big), \sigma_u\big(t, \Psi_d(t)\big)\Big), \tag{36}$$

where the normal distribution mean μ_u and standard deviation σ_u are based on that of the multi-level ensemble using Eq. (11) for the velocity u at time t and position $\Psi_d(t)$ for drifters $d = 1, ..., N_d$. A corresponding expression is used for v.

If the sampling in Eq. (36) is uncorrelated in time, however, each ensemble member representing a drifter will be an estimate of the mean trajectory as t increases, and the ensemble spread will be underestimated. This can be seen through a comparison to the standard single-level "one drifter per ocean model" approach. In the standard case, a drifter that is associated with an ocean state that represents the tail of the ocean state probability distribution will consistently generate a corresponding trajectory that will represent the tail of the trajectory distribution as well. By sampling the currents according to Eq. (36), the probability that one trajectory consistently follows the tail of the distribution for the current is highly unlikely.

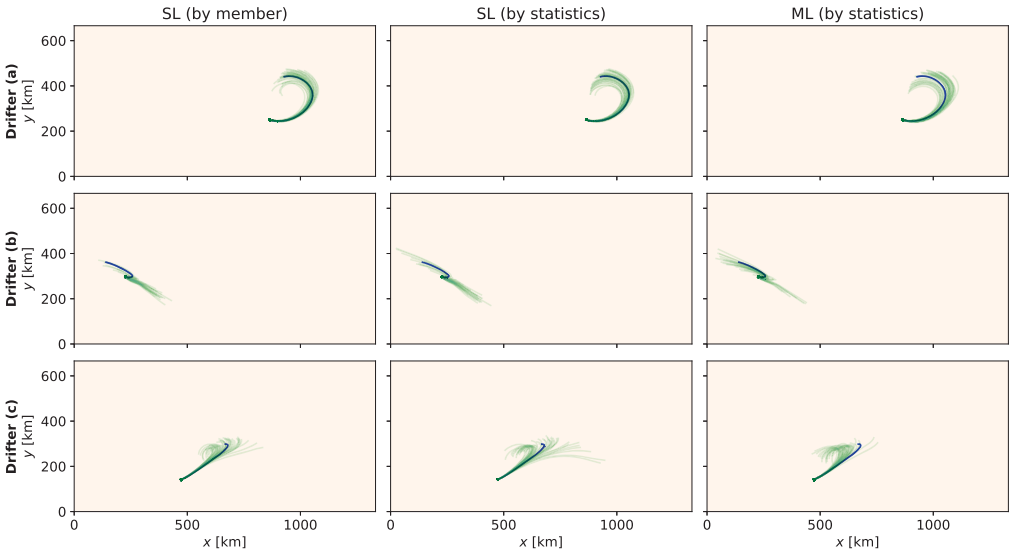


Figure 10: Drift trajectories for single-level and multi-level ensembles with different trajectory modelling approaches. We use 50 ensemble members in the forecasts by statistics, and for the multi-level plot, we use the ensemble of case (A). The blue trajectory comes from the nature run. We show the trajectories for three different starting positions.

For this reason, we instead simulate drifters through what can be described as a *biased walk*. Thus, we let $\beta_{d,u}, \beta_{d,v} \sim \mathcal{N}(0,1)$ be a random normal distributed bias for the velocities u and v, respectively. We sample these only once for each drifter realisation $d = 1, ..., N_d$ and keep these fixed over time. We then evolve each drifter's position by

$$\Psi_d(t + \Delta t) = \Psi_d(t) + \Delta t \begin{bmatrix} \mu_u(t, \Psi_d) + \beta_{d,u}\sqrt{\sigma_u(t, \Psi_d(t))} \\ \mu_v(t, \Psi_d) + \beta_{d,v}\sqrt{\sigma_v(t, \Psi_d(t))} \end{bmatrix}, \tag{37}$$

so that each drifter follows a consistent path within the probability distribution for the currents. Note that Eq. (37) also can be used for a single-level ensemble by using the corresponding single-level estimators for the velocities.

6.2 Numerical drift results

In the following, we generate three days trajectory forecasts for passive drifters starting at 100 randomly chosen initial locations in the domain at day 10 of our experiment as described in Sect. 5.1.4. We consider three different cases: Classical single-level trajectories generated by attaching drifters to each ocean model ensemble member; using the statistical approach of the biased walk in Eq. (37) for exactly the same single-level run; and using the biased walk for the multi-level ensemble experiment from case (A) in Sect. 5. For the biased walks, we choose a drifter ensemble size of $N_d = 50$ and $\Delta t = 60$ s as the drifter time step.

Figure 10 shows trajectory forecasts using the three different methods for three different initial positions. We compare them with the drift trajectories generated by the nature run, here referred to as the *true drifters/trajectories*. Note that these three initial positions are hand picked from the set of all initial positions, with the intention of showcasing typical forecasts and give examples of the multi-level forecast being both

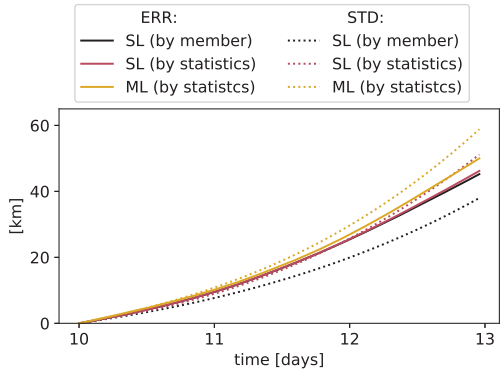


Figure 11: Error in the mean and standard deviation of the drift trajectory forecasts.

better and worse than the single-level forecast. For drifter (a) in the top row, all forecast trajectories follow the path of the true trajectory and cover the true drifter's final position with slightly different spread. Drifter (b) is harder to predict as all three methods suggest multi-model distributions that indicates that it can either drift down to the right or up to the left. We see, however, that the multi-level ensemble suggests higher probability to the left which is correct, whereas both methods based on the single-level ensemble give higher probability to the right. On the other side, the true trajectory for drifter (c) in the lowermost row is not well represented in the multi-level ensemble. Here, the single-level ensemble gives a good prediction of the true trajectory when we make the forecast directly with one drifter per ocean ensemble member. For the biased walk with the same single-level ensemble, the true trajectory is still well represented by the ensemble, but the ensemble has a larger spread.

This shows that it is difficult to assess the quality of each method based on the trajectory forecast of just a few initial positions. We therefore study the error in the mean for all 100 initial positions, similar to what was done for the state estimation. During the forecast period in the experiment, the error in the mean for the drifter position at time t is

$$\mathrm{ERR}(t) = \|\mu[\Psi(t)] - \Psi_{\mathrm{true}}(t)\|, \tag{38}$$

where $\|\cdot\|$ is the l^2-vector norm.

In Fig. 11, the drifter error ERR and the standard deviation of the drifter ensemble (STD) is averaged over all 100 initial positions and plotted over time for the different methods. The figure shows that the results for the error in the mean are similar for all methods, but slightly larger during the final forecast day for the multi-level ensemble. Considering the standard deviations, it is clear that the single-level ensemble generates a smaller ensemble spread for the trajectories than the multi-level ensemble. This is in line with the results of the state estimation in Fig. 6. The displays also show that the spread of the biased walk is larger than the spread of the traditional method per ocean state for the single-level ensemble. Since the same single-level ocean model ensemble is used for both trajectory modelling approaches here, these differences do not originate in the ocean states but purely from the drifter modelling. Hence, we can pinpoint the cause of the increased spread of the multi-level drifter ensemble compared to the classical single-level approach with one drifter per member, as first being due to the biased walk modelling and second due to the ensemble spread in the ocean states.

We have repeated the experiments with varying parameters for the biased walk and observed that for $N_d < 50$ drifters, results are not robust, while for $N_d > 50$ the improvements start to stagnate. Similarly, for drifter time steps up to approximately $\Delta t = 5$ min the results are qualitatively comparable.

From the final positions of the modelled drifters, the probabilities of their where-about can be calculated by kernel density estimation. Figure 12 shows the areas corresponding to the probability levels of 90, 75, 50

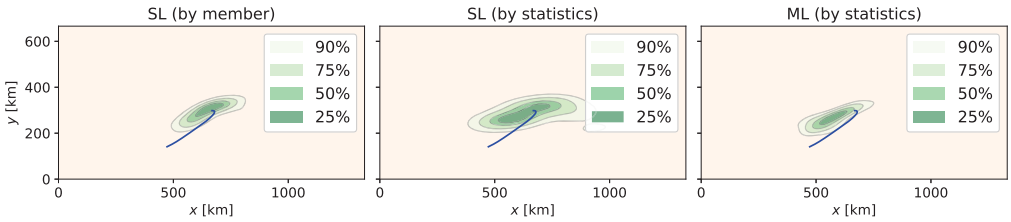


Figure 12: Levels of estimated kernel densities for the final position of drifter (c) from Figure 10. True trajectory is plotted in blue.

Table 2: Probabilities of the true drifters to be in the corresponding probability zone at the end of the forecast.

Drifter Case	**Probability levels** 25%	50%	75%	90%
SL (by member)	16	46	69	85
SL (by statistics)	26	55	74	90
ML (by statistics)	38	65	84	91

and 25% for drifter (c), along with its true trajectory in blue. The mode and the spread of the probability distributions differ between the methods, but as in Fig. 10, the single-level ensembles cover this particular drifter slightly better than the multi-level ensemble.

To investigate coverage properties further, we follow the idea of rank histograms, and record the number of times a drifter ends in each of the four probability levels. Table 2 shows the results based on 100 simulated true drifters, using the same probability levels as in Fig. 12. The counting results suggest that the single-level with biased walk has the smallest mismatch with theoretical probabilities. The classical single-level is slightly under-dispersive and the multi-level trajectories are slightly over-dispersive. This is in line with the results from Figure 11, where the error in the mean is the same for all three methods, but the spread in the ensembles differ correspondingly at day 13.

7 Concluding Remarks

In this article, we have applied the MLEnKF successfully to a simplified ocean model. The goal of multi-level methods is to gain a computational speed-up compared to a single-level method, while maintaining the same statistical accuracy. While MLDA has shown potential in a theoretical context and in analytical examples, we elevate its applicability one step closer to practical relevance in ocean forecasting.

In Sect. 3, we revised the MLEnKF by Chernov et al. (2021) for nested grids in finite-volume methods. We discussed practical challenges that arise when implementing MLDA. These considerations include the choice of multi-level ensemble sizes, handling localisation, as well as the mitigation of negative eigenvalues in the covariance estimation. Details are discussed in Sect. 4. Conclusions relating to each of the research questions from Sect. 1 are as follows:

i) We re-used our existing GPU-accelerated simulator GPU Ocean, which is optimised for running single-level ensembles of high-resolution simplified ocean models, and extended it by the integration of the MLEnKF. Although a flexible choice of the resolution in the simulator enables one to run ensembles on different levels, an efficient choice of the model error becomes crucial for multi-level ensembles with coupled pairs.

As multi-level methods profit from lower computational costs on coarser levels, the performance scaling between the levels directly influences the speed-up. Even though the GPU framework in our example facilitates very fast simulations, it has suboptimal performance scaling for problems with very coarse grid resolutions. This reduced the potential of computational benefits through MLDA. Meanwhile, we were still interested in the upper bound of the computational speed-up, that may be obtainable for other software frameworks. We therefore discussed speed-up under assumptions of both theoretical and practical work. We obtained a theoretical speed-up up to a factor 2.

ii) The speed-up depends on both the quantity of interest and the corresponding variance scaling between the levels. We ran a trial experiment with single-level data assimilation to record the variances per level at the final time and discussed two quantities of interest that are purposeful for data assimilation.

For a large speed-up, strong correlations between the fine levels are favourable. However, if the simulation results on the finer levels become too similar to each other, there would be no incentive to use the finest resolutions in practice as they would not provide any new information. In such a case, defects in the forward step often lie in the model and are not due to the resolution. Nevertheless, there may be external reasons to keep very fine resolutions.

While MLMC is designed for a single estimation, multi-level ensembles are re-used for multiple estimations in the MLDA framework. For practical reasons, we used one multi-level ensemble for all estimations throughout the full numerical experiments. The multi-level configuration influenced the theoretical speed-up, but we have seen that it does not change the quality of the state estimation and forecast in any relevant range.

iii) To assess the quality of MLDA, we tested three different multi-level ensemble configurations, matching the theoretical statistical accuracy of an estimation with a given single-level ensemble. All three cases resulted in qualitatively the same mean and spread compared to each other. Rank histograms from repeated short-span experiments indicated an appropriate fit of the ensemble distribution.

Compared to the single-level experiment, the multi-level experiments gave similar error in the mean but slightly larger spread for the partially observed momentum, but significantly lower error and spread for the unobserved sea-surface elevation.

iv) We have introduced a drift model based on a so-called biased walk, that relies on estimates of the mean and the variance of the currents. Using this method, we demonstrated how the posterior multi-level ensemble can be used to forecast drift trajectories. We compared the classical single-level drifter modelling with one drifter per ensemble member to the biased walk approach, where we used both the single-level and the multi-level ensemble for the estimation. The averaged results for 100 initial drifter positions show that the error in the mean is qualitatively the same for all, but the spreads of the drifter ensemble differ.

In this article, we have laid the groundwork for a general discussion of the applicability of MLDA in ocean forecasting and we have addressed the most prominent points required to advance multi-level methods further towards practical relevance in data assimilation. The next steps required for even further advancements of MLDA can be seen in several independent directions. One is to develop and study MLEnKF variants that are suitable for assimilating larger amounts of observational data. Another direction is to carry out variance analysis in real-world domains to assess the potential computational speed-up that can be expected in real-world cases. In addition, there are more practical challenges related to real-world domains that need to be handled to successfully apply multi-level methods, such as mismatching land masks between levels and consistent coarsening of bottom topography.

Code Availability The source code used to produce the results presented in this paper is available under a GNU free and open source license in order to enhance scientific exchange, see `https://github.com/metno/gpuocean` for the core and `https://github.com/FlorianBeiser/multilevelDA` for the examples.

Acknowledgements This work is part of the Havvarsel project supported by the Research Council of Norway under grant number 310515.

References

Bannister, R. N. (2017). A review of operational methods of variational and ensemble-variational data assimilation. *Quarterly Journal of the Royal Meteorological Society*, 143(703):607–633.

Beiser, F., Holm, H. H., and Eidsvik, J. (2024). Comparison of ensemble-based data assimilation methods for sparse oceanographic data. *Quarterly Journal of the Royal Meteorological Society*, 150(759):1068–1095.

Beskos, A., Jasra, A., Law, K., Tempone, R., and Zhou, Y. (2017). Multilevel sequential monte carlo samplers. *Stochastic Processes and their Applications*, 127(5):1417–1440.

Bierig, C. and Chernov, A. (2014). Convergence analysis of multilevel monte carlo variance estimators and application for random obstacle problems. *Numerische Mathematik*, 130(4):579–613.

Bierig, C. and Chernov, A. (2016). Approximation of probability density functions by the multilevel monte carlo maximum entropy method. *Journal of Computational Physics*, 314:661–681.

Bishop, C. H., Etherton, B. J., and Majumdar, S. J. (2001). Adaptive sampling with the ensemble transform Kalman filter. Part I: Theoretical aspects. *Monthly weather review*, 129(3):420–436.

Brodtkorb, A. and Holm, H. (2021). Coastal ocean forecasting on the GPU using a two-dimensional finite-volume scheme. *Tellus, Series A: Dynamic Meteorology and Oceanography*, 73(1):1–22.

Carrassi, A., Bocquet, M., Bertino, L., and Evensen, G. (2018). Data assimilation in the geosciences: An overview of methods, issues, and perspectives. *Wiley Interdisciplinary Reviews: Climate Change*, 9(5):e535.

Chen, Y. and Oliver, D. S. (2017). Localization and regularization for iterative ensemble smoothers. *Computational Geosciences*, 21(1):13–30.

Chernov, A., Hoel, H., Law, K., Nobile, F., and Tempone, R. (2021). Multilevel ensemble Kalman filtering for spatio-temporal processes. *Numerische Mathematik*, 147(1):71–125.

Chertock, A., Dudzinski, M., Kurganov, A., and Lukácová-Medvidová, M. (2017). Well-balanced schemes for the shallow water equations with Coriolis forces. *Numerische Mathematik*.

Chiodaroli, E., De Lellis, C., and Kreml, O. (2015). Global ill-posedness of the isentropic system of gas dynamics. *Communications on Pure and Applied Mathematics*, 68(7):1157–1190.

Dagestad, K.-F., Röhrs, J., Breivik, Ø., and Ådlandsvik, B. (2018). OpenDrift v1.0: a generic framework for trajectory modelling. *Geoscientific Model Development*, 11(4):1405–1420.

Delandmeter, P. and van Sebille, E. (2019). The Parcels v2.0 Lagrangian framework: new field interpolation schemes. *Geoscientific Model Development*, 12(8):3571–3584.

Destouches, M., Mycek, P., and Gürol, S. (2023). Multivariate extensions of the multilevel best linear unbiased estimator for ensemble-variational data assimilation. *arXiv preprint arXiv:2306.07017*.

Dodwell, T., Ketelsen, C., Scheichl, R., and Teckentrup, A. (2015). A hierarchical multilevel markov chain monte carlo algorithm with applications to uncertainty quantification in subsurface flow. *SIAM/ASA Journal on Uncertainty Quantification*, 3(1):1075–1108.

Evensen, G. (2006). *Data Assimilation: The Ensemble Kalman Filter*. Springer Berlin Heidelberg.

Evensen, G., Vossepoel, F., and van Leeuwen, P. (2022). *Data assimilation fundamentals: A unified formulation of the state and parameter estimation problem*. Springer Nature.

Farmer, C. (2002). Upscaling: a review. *International Journal for Numerical Methods in Fluids*, 40(1-2):63–78.

Fjordholm, U. S., Lye, K., Mishra, S., and Weber, F. (2020). Statistical solutions of hyperbolic systems of conservation laws: Numerical approximation. *Mathematical Models and Methods in Applied Sciences*, 30(03):539–609.

Fossum, K., Mannseth, T., and Stordal, A. (2020). Assessment of multilevel ensemble-based data assimilation for reservoir history matching. *Computational Geosciences*, 24(1):217–239.

Galewsky, J., Scott, R., and Polvani, L. (2004). An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):429–440.

Giles, M. (2008). Multilevel monte carlo path simulation. *Operations Research*, 56(3):607–617.

Giles, M. (2015). Multilevel Monte Carlo methods. *Acta Numerica*, 24:259–328.

Gneiting, T., Balabdaoui, F., and Raftery, A. (2007). Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 69(2):243–268.

Gregory, A. and Cotter, C. (2017). On the calibration of multilevel monte carlo ensemble forecasts. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1929–1935.

Hagen, T. R., Henriksen, M. O., Hjelmervik, J. M., and Lie, K.-A. (2007). How to solve systems of conservation laws numerically using the graphics processor as a high-performance computational engine. *Geometric Modelling, Numerical Simulation, and Optimization: Applied Mathematics at SINTEF*, pages 211–264.

Hamill, T. (2001). Interpretation of rank histograms for verifying ensemble forecasts. *Monthly Weather Review*, 129(3):550–560.

Heinrich, S. (2001). Multilevel monte carlo methods. In *Large-Scale Scientific Computing: Third International Conference, LSSC 2001 Sozopol, Bulgaria, June 6–10, 2001 Revised Papers 3*, pages 58–67. Springer.

Hoel, H., Law, K., and Tempone, R. (2016). Multilevel ensemble kalman filtering. *SIAM Journal on Numerical Analysis*, 54(3):1813–1839.

Hoel, H., Shaimerdenova, G., and Tempone, R. (2020). Multilevel ensemble kalman filtering with local-level kalman gains. *arXiv preprint arXiv:2002.00480*.

Holm, H., Sætra, M., and Brodtkorb, A. (2020a). Data assimilation for ocean drift trajectories using massive ensembles and GPUs. In Klöfkorn, R., Keilegavlen, E., Radu, F., and Fuhrmann, J., editors, *Finite Volumes for Complex Applications IX - Methods, Theoretical Aspects, Examples*, pages 715–723, Cham. Springer International Publishing.

Holm, H., Sætra, M., and van Leeuwen, P. (2020b). Massively parallel implicit equal-weights particle filter for ocean drift trajectory forecasting. *Journal of Computational Physics: X*, 6(0314):100053.

Hunt, B. R., Kostelich, E. J., and Szunyogh, I. (2007). Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter. *Physica D: Nonlinear Phenomena*, 230(1-2):112–126.

Jasra, A., Kamatani, K., Law, K., and Zhou, Y. (2017). Multilevel particle filters. *SIAM Journal on Numerical Analysis*, 55(6):3068–3096.

Jasra, A., Law, K., and Suciu, C. (2020). Advanced Multilevel Monte Carlo Methods. *International Statistical Review*, 88(3):548–579.

Latz, J., Papaioannou, I., and Ullmann, E. (2018). Multilevel sequential^2 monte carlo for bayesian inverse problems. *Journal of Computational Physics*, 368:154–178.

Lye, K. (2019). Alsvinn: A fast multi-gpgpu finite volume solver with a strong emphasis on reproducibility. *arXiv preprint arXiv:1912.07645*.

Lye, K. (2020). *Computation of statistical solutions of hyperbolic systems of conservation laws*. PhD thesis, ETH Zurich.

Maurais, A., Alsup, T., Peherstorfer, B., and Marzouk, Y. (2023). Multifidelity covariance estimation via regression on the manifold of symmetric positive definite matrices. *arXiv preprint arXiv:2307.12438*.

Mishra, S. and Schwab, C. (2012). Sparse tensor multi-level monte carlo finite volume methods for hyperbolic conservation laws with random initial data. *Mathematics of Computation*, 81(280):1979–2018.

Mishra, S., Schwab, C., and Šukys, J. (2012a). Multi-level monte carlo finite volume methods for nonlinear systems of conservation laws in multi-dimensions. *Journal of Computational Physics*, 231(8):3365–3388.

Mishra, S., Schwab, C., and Šukys, J. (2012b). Multilevel monte carlo finite volume methods for shallow water equations with uncertain topography in multi-dimensions. *SIAM Journal on Scientific Computing*, 34(6):B761–B784.

Müller, F. (2014). *Stochastic methods for uncertainty quantification in subsurface flow and transport problems*. PhD thesis, ETH Zurich.

Mycek, P. and De Lozzo, M. (2019). Multilevel monte carlo covariance estimation for the computation of sobol' indices. *SIAM/ASA Journal on Uncertainty Quantification*, 7(4):1323–1348.

Nezhadali, M., Bhakta, T., Fossum, K., and Mannseth, T. (2020). A Novel Approach to Multilevel Data Assimilation. In *Conference Proceedings, ECMOR XVII*, pages 1 – 13. European Association of Geoscientists & Engineers.

Rabault, J., Nose, T., Hope, G., Müller, M., Breivik, Ø., Voermans, J., Hole, L. R., Bohlinger, P., Waseda, T., Kodaira, T., Katsuno, T., Johnson, M., Sutherland, G., Johansson, M., Christensen, K. H., Garbo, A., Jensen, A., Gundersen, O., Marchenko, A., and Babanin, A. (2022). OpenMetBuoy-v2021: An easy-to-build, affordable, customizable, open-source instrument for oceanographic measurements of drift and waves in sea ice and the open ocean. *Geosciences*, 12(3).

Røed, L. (2012). Documentation of simple ocean models for use in ensemble predictions. Part I: Theory. Technical report, Norwegian Meteorological Institute.

Sakov, P. and Bertino, L. (2010). Relation between two common localisation methods for the EnKF. *Computational Geosciences*, 15(2):225–237.

Schaden, D. and Ullmann, E. (2020). On multilevel best linear unbiased estimators. *SIAM/ASA Journal on Uncertainty Quantification*, 8(2):601–635.

Shchepetkin, A. and McWilliams, J. (2005). The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modelling*, 9(4):347–404.

Shivanand, S. K. (2023). Covariance estimation using h-statistics in monte carlo and multilevel monte carlo methods. *arXiv preprint arXiv:2311.01336*.

Sun, L. and Penny, S. (2019). Lagrangian data assimilation of surface drifters in a double-gyre ocean model using the local ensemble transform kalman filter. *Monthly Weather Review*, 147(12):4533–4551.

Van Leeuwen, P., Künsch, H., Nerger, L., Potthast, R., and Reich, S. (2019). Particle filters for high-dimensional geoscience applications: A review. *Quarterly Journal of the Royal Meteorological Society*, 145(723):2335–2365.

Vetra-Carvalho, S., van Leeuwen, P., Nerger, L., Barth, A., Altaf, M., Brasseur, P., Kirchgessner, P., and Beckers, J.-M. (2018). State-of-the-art stochastic data assimilation methods for high-dimensional non-Gaussian problems. *Tellus A: Dynamic Meteorology and Oceanography*, 70(1):1–43.

Zhang, F., Snyder, C., and Sun, J. (2004). Impacts of initial estimate and observation availability on convective-scale data assimilation with an ensemble kalman filter. *Monthly Weather Review*, 132(5):1238–1253.

A Initialisation and turbulence in the numerical experiments

The initial state used in the numerical experiments in Sect. 5 is inspired by Galewsky et al. (2004), who proposed a test case for validating numerical solvers for shallow-water models on a rotating sphere. This test case was originally stated for atmospheric flows, describing an eastward jet centred at 45° north kept in geostrophic balance due to the sphere's rotation and the fluid's thickness, which takes the form of a smooth step function. Although the initial state is in a steady state, the introduction of a small perturbation in the fluid depth gives rise to instabilities in the jet, which after some time develops to complex currents and eddies. By placing these perturbations at random, e.g., by adding model errors, the instabilities develop in different ways and at different times, thereby motivating the use of such a test case for identical-twin data assimilation experiments.

Holm et al. (2020b) adapted this case to oceanic conditions and transformed it to a rectangular flat domain with a constant Coriolis parameter and periodic boundary conditions. To ensure compatible conditions at the northern and southern boundary, they introduced a second westward jet, similar but opposite to the original one. The domain was modelled after the conditions in the Barents sea, using a Coriolis parameter $f = 1.405 \cdot 10^{-4} \frac{\text{m}}{\text{s}}$ (corresponding to 75° north) and $H = 228$ m. The jet itself was given an initial maximum value of 3 $\frac{\text{m}}{\text{s}}$ and a width corresponding to 222 km (about 2° of latitude), whereas the length of the domain along y was 666 km.

The original initial conditions for the velocity of the jet u and the corresponding depth of the fluid h are given as analytically as function of the latitude (see Eqs. (2) and (3) in Galewsky et al. (2004)), where h requires that we solve a numerical integral involving u and u^2. To map these to the flat rectangular domain, the analytic expression for u was first used to find an approximate value for h, which was then used to reconstruct u using Eq. (31). This recalculation of u was crucial for obtaining a fully balanced initial state. Figure 13 shows a cross section along y of the resulting initial conditions for the sea-surface deviation η and the velocity u in x-direction. Due to way these initial conditions are constructed, they are not available in analytical expressions, but the simulation framework GPU Ocean contains code to generate the initial state for the domain 1332 km × 666 km with any grid resolution. An example can be found in the Jupyter Notebook `notebooks/DoubleJet-IC.ipynb` in `https://github.com/FlorianBeiser/multilevelDA`.

To illustrate the turbulence arising when simulating this state using the model error described in Sect. 5.1.2, we run a pure single-level forecast experiment using 100 ensemble members on a 512 × 265 grid resolution. Figures 14 and 15 show the standard deviation in hu and hv, respectively, at four different points in time. From an initial spread of zero, the turbulent currents start evolving first slightly at day three and then more prominent at day five. This is shown by an ensemble spread at the edges of the jets for hu, and in the middle of the jets for hv. Later, the turbulence develop complex currents and eddies that give a spread throughout the entire domain. Finally, Fig. 16 shows how the norm of the standard deviation develops as a function of time. Note specifically how the ensemble spread increases most rapidly between days 3 to 6, before increasing at a lower rate throughout the rest of the experimental period. We do note that a simplified shallow-water

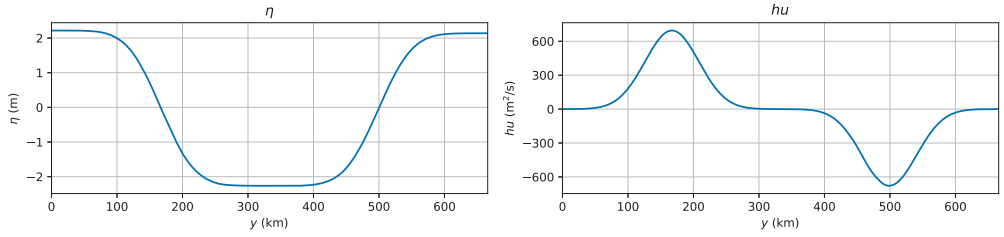


Figure 13: Initial state of the numerical experiments, shown here as a cross-section along y. This figure is previously published in Holm et al. (2020b).

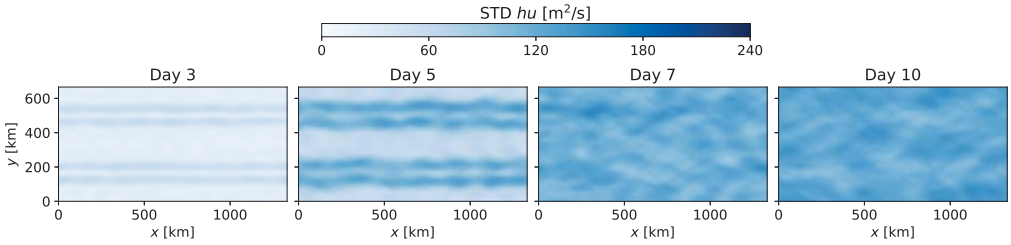


Figure 14: STD of hu throughout the domain without data assimilation.

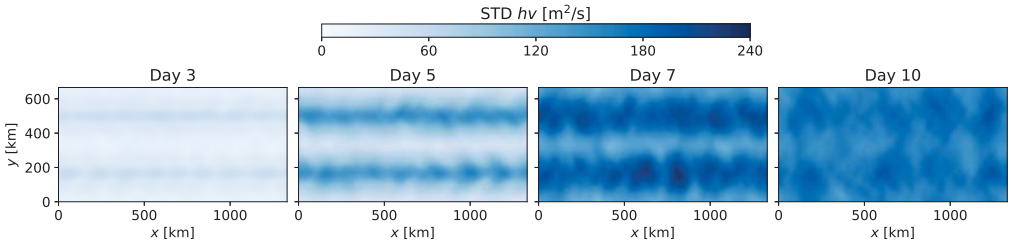


Figure 15: STD of hv throughout the domain without data assimilation.

system (equivalent to the isentropic 2D Euler equations) are known to be globally ill-defined Chiodaroli et al. (2015), exhibiting so-called "wild" or "surprising" solutions. While the ill-posedness does not directly come into play in the mentioned initial data, we do note that the ill-posedness of these equations are intrinsically linked to turbulence Lye (2020).

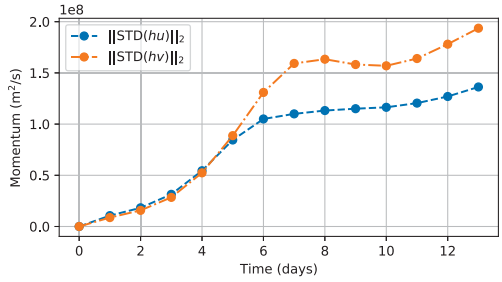


Figure 16: The L2-norm of STD for hu and hv as a function of time.