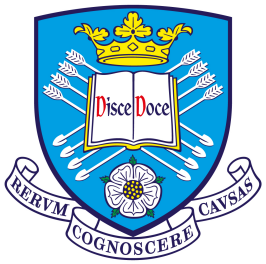


The University Of Sheffield.

Autonomous Monitoring of Contaminants in Fluids

Submitted September 2020, in partial fulfillment of
the conditions for the award of the degree **PhD - Automatic Control and Systems Engineering.**

Zak Hodgson
160120712

Supervised by Dr Bryn Jones

Department of Automatic Control and Systems Engineering
University of Sheffield

I hereby declare that this dissertation is all my own work, except as indicated in the text:

To my Grandad John, who has always wanted a Doctor in the family,
and to my dog Gandalf, who would happily eat this thesis.

"I may not have gone where I intended to go,
but I think I have ended up where I needed to be." - Douglas Adams, 1952 - 2001.

Acknowledgements

This thesis is the culmination of 4 years of work, work that was at times infuriating, but also incredibly satisfying. The PhD started from a chance conversation with Professor Tony Dodd, then Dr Bryn Jones and then with James and David Browne of Andrew-Moore and Associates (AMA). With their support, knowledge and ample amounts of paperwork, the next 4 years of my academic life were arranged and I owe them all great gratitude for getting the ball rolling and arranging funding. Further thanks should go to my principal supervisors, Dr Bryn Jones and David Browne. Both have never been more than a few words away, with academic and industrial opportunity, ideas and guidance. I am indebted to Bryn for, on-top of all the valuable supervisory work and administration, he has spent 4 years listening to my breathless rants and reading my far-too-long "Zak sentences". Within the University of Sheffield, I would also like to thank Iñaki Esnaola for taking on the role of second supervisor, and his extremely helpful and rigorous input. Also, Matthew Ham for his administrative legwork and Darren Fox for his equipment procurement.

To my friends, Billy, Shiv and Oliver, thank you for your support, proof-reading and sanity checks. To my old friend Jon, you have been a voice of reason and consultation about the wider world. To my friend of 8 years and house-mate for 6, Dan, a heart-felt thank you for the adventures, equal part enjoyable and exhausting, we have had. I would either be absorbed into my desk, or stranded somewhere in the Alps without you. I look forward to more adventures.

To Lily, thank you for your unending faith in my ability, and your home-improvements, patience and understanding. I suspect I owe my good health to you urging me to take a break and get out of the house. My apologies for spending far too much time staring vacantly into the distance on walks, obsessing over work, rather than enjoying your company. Things will get better.

Lastly, to my family. To my Uncle Shaun and Grandma Rosemary, thank you for your support and never being further than a phone call away. To my parents, Jill and Jeremy, you have made me who I am: Completing this PhD is a reflection of the strength, education and upbringing you gave me, thank you, and thank you for always being there.

Related Publications

Much of the work contained in this thesis is also available in the following publications:

- Zak Hodgson et al. (2019). "A combined ocean and oil model for model-based adaptive monitoring". In: arXiv: `1910.12658` - Also submitted to UKACC 2020, but postponed indefinitely due to COVID-19.

- Zak Hodgson, Iñaki Esnaola, and Bryn Jones (2020b). "Optimal Model-Based Sensor Placement & Adaptive Monitoring Of An Oil Spill". In: *IFAC-V*.

- Zak Hodgson, Iñaki Esnaola, and Bryn Jones (2020a). "Model-Based Optimal Adaptive Monitoring of Oil". In: *IEEE Transactions on Control Systems Technology - Submitted*.

Abstract

The litigation and mitigation of maritime incidents suffer from a lack of information, first at the incident location, then throughout the evolution of contaminants such as spilled oil through the surrounding environment. Prior work addresses this through ocean and oil models, model directed sensor guidance and other observation methods such as satellites. However, each of these approaches and research fields have short-comings when viewed in the context of fast-response to an incident, and of constructing an all-in-one framework for monitoring contaminants using autonomous mobile sensors. In summary, models often lack consideration of data-assimilation or sensor guidance requirements, sensor guidance is specific to source locating, oil mapping, or fluid measuring and not all three, and data assimilation methods can have stringent requirements on model structure or computation time that may not be feasible.

This thesis presents a model-based adaptive monitoring framework for the estimation of oil spills using mobile sensors. In the first of a four-stage process, simulation of a combined ocean, wind and oil model provides a state trajectory over a finite time horizon, used in the second stage to solve an adjoint optimisation problem for sensing locations. In the third stage, a reduced-order model is identified from the state trajectory, utilised alongside measurements to produce smoothed state estimates in the fourth stage, which update and re-initialise the first-stage simulation. In the second stage, sensors are directed to optimal sensing locations via the solution of a Partial Differential Equation (PDE) constrained optimisation problem. This problem formulation represents a key contributory idea, utilising the definition of spill uncertainty as a scalar PDE to be minimised subject to sensor, ocean, wind and oil constraints. Spill uncertainty is a function of uncertainty in (i) the bespoke model of the ocean, wind and oil spill, (ii) the reduced order model identified from sensor data, and (iii) the data assimilation method employed to estimate the states of the environment and spill. The uncertainty minimisation is spatio-temporally weighted by a function of spill probability and information utility, prioritising critical measurements.

In the penultimate chapter, numerical case-studies spanning a 2500 km^2 coastal area are presented. Here the monitoring framework is compared to an industry standard method in three scenarios: A spill monitoring and prediction problem, a retrodiction and monitoring problem and a source locating problem.

Abbreviations

DAE Differential Algebraic Equation.

DMD Dynamic Mode Decomposition.

GNOME General NOAA Operational Modeling Environment.

ITOPF The International Tanker Owners Pollution Federation.

KF Kalman Filter.

PDE Partial Differential Equation.

RTS Rauch–Tung–Striebel.

SAR Synthetic Aperture Radar.

SCEM Sheffield Combined Environment Model.

SLAR Side-Looking Airborne Radar.

SWEM Sheffield Wave Environment Model.

UAV Unmanned-Airborne-Vehicle.

USV Unmanned-Submersible/Seaborne-Vehicle.

Nomenclature

Common terms only. Single use and locally used terms are omitted.

α_{c_0} Coefficient that attenuates the velocity of oil particles resulting from ocean velocity.

α_{w_0} Coefficient that attenuates the velocity of oil particles resulting from wind velocity.

$\bar{\mathscr{H}}$ The Hankel data matrix formed of centred state trajectories, excluding the last column.

$\bar{\mathscr{H}}'$ The Hankel data matrix formed of centred state trajectories, excluding the first column.

$\bar{\mathscr{H}}_r$ The Hankel data matrix formed of centred state trajectories.

$\bar{\mathscr{K}}$ Column vector trajectory of centred external data.

$\bar{\mathscr{Y}}$ Column vector trajectory of centred sensor measurements.

$\boldsymbol{\Phi}$ The mode shape matrix, or linear transformation matrix, that transforms the reduced order state vector to a trajectory of full order states, contained in a single column of the Hankel data matrix.

$\boldsymbol{\Psi}$ Reduced order model mapping from reduced order states to current-time full order states.

$\boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})$ Uncertainty state space system state transition matrix.

$\boldsymbol{C}$ Sensor/output matrix for a sensor measurement and external data trajectory.

$\boldsymbol{C}_{\mathrm{KF}}$ Sensor/output matrix for current-time sensor measurements only.

$\boldsymbol{F}(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})$ The state trajectory of the uncertainty state space system, as constructed by a particular implementation (4.111).

$\boldsymbol{L}$ Kalman filter gain.

$\boldsymbol{P}$ Full order model estimated error covariance matrix.

$\boldsymbol{P}_{\mathrm{r}}$ Reduced order model estimated error covariance matrix.

Δt The regular time-step in the reduced order model.

δt The time-varying time-step in the temporal discretisation.

δt_k The time-varying time-step in the temporal discretisation, at time-step index k.

δx Grid spacing in the horizontal direction.

δy Grid spacing in the vertical direction.

δz_1 Fine grid spacing in the water depth discretisation.

δz_2 Coarse grid spacing in the water depth discretisation.

ϵ_x The error covariance of external data flow velocity in the horizontal direction.

ϵ_y The error covariance of external data flow velocity in the vertical direction.

κ Wind resistance coefficient.

$\Lambda^{\frac{\delta t}{\Delta t}}$ Reduced order state transition matrix from one time-step to the next.

$\mathbb{C}$ The set of complex numbers.

$\mathbb{N}$ The set of natural numbers.

$\mathbf{z}(x_i, y_i)$ The set of depths at grid-cell i, j.

$\mathscr{F}(\overset{u}{\underset{\sim}{\vec{x}}}_{k+1}, \overset{u}{\underset{\sim}{\vec{x}}}_{k}, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})$ Differential Algebraic Equation for a time-step of the uncertainty state space system.

ν Fluid kinematic viscosity.

Ω 3D spatial domain.

$\overset{u}{\vec{\mathscr{X}}}$ State trajectory of the uncertainty state space system.

$\overset{u}{\vec{x}}$ State vector of the uncertainty state space system.

$\partial\Omega$ 2D spatial domain at the air/water interface.

$\mathbb{R}$ The set of real numbers.

$\mathbb{R}_+$ The set of positive real numbers, excluding zero.

$\rho_{(\cdot)}$ The density of $(\cdot)$ unless otherwise stated.

$\sigma_{\cdot}$ The standard deviation of $(\cdot)$, unless otherwise stated.

$\sigma_{\cdot}^2$ The variance of $(\cdot)$.

σ_x^2 The variance in particle position, in the horizontal direction.

σ_y^2 The variance in particle position, in the vertical direction.

$\underline{H}_r(\cdot)$ Limited Heaviside function that activates sensor uncertainty removal in a radius around the sensor.

$\underline{H}_t(\cdot)$ Limited Heaviside function that activates sensor uncertainty removal after a travel time.

$\vec{\mathscr{P}}$ Sensor position trajectory.

$\vec{\mathscr{X}}$ State trajectory of the combined environment and oil model.

$\vec{p}$ Row vector of sensor positions.

$\vec{s}_U$ External forces acting upon a flow.

$\vec{U}$ Flow velocity, or oil advection velocity.

$\vec{U}_{\mathrm{c}}$ Ocean current flow velocity.

$\vec{U}_{\mathrm{E}}$ Ekman wind flow velocity.

$\vec{U}_{\mathrm{w}}$ Wind flow velocity.

$\vec{U}_s$ Stokes drift, wave induced velocity.

$\vec{x}$ State vector of the combined environment and oil model.

$\vec{x}$ The Sheffield Combined Environment Model state vector.

$\vec{z}$ State vector of the reduced order system.

$c(\vec{x}, t, \vec{p})$ The sensor constraint function.

D_h The horizontal diffusivity coefficient.

D_{v_z} The vertical diffusivity coefficient.

$E(\vec{x}, t, \vec{p})$ Spatio-temporal weighting of uncertainty in the sensor pathing cost function.

$E_{\mathrm{KF}_x}(\vec{p})$ The data assimilation and reduced order modelling error for velocity in the horizontal direction.

$E_{\mathrm{KF}_y}(\vec{p})$ The data assimilation and reduced order modelling error for velocity in the vertical direction.

g Gravitational constant

h_c Number of columns in the Hankel data matrix.

h_r Number of rows in the Hankel data matrix.

i, j, w The horizontal, vertical and depth grid indices respectively, unless otherwise stated for a particular equation.

J Sensor pathing optimisation cost function evaluation.

k The discrete time-step index in the time interval $[t_0, t_f]$.

k_σ Number of standard deviations used in a confidence interval.

n_{cell} The number of states per grid-cell in the Sheffield Combined Environment Model.

n_ϕ Number of pressure field correction iterations.

n_τ Number of discrete time-steps in sensor pathing optimisation.

N_{crit} Number of fine mesh grid cells in the water depth discretisation.

n_k Number of external data states at a given time.

n_p Number of sensors.

n_r Number of repetitions of the state trajectory used in the Dynamic Mode Decomposition.

n_s The total number of states in the Sheffield Combined Environment Model.

n_t Number of time-lagged state trajectories in the Hankel data matrix.

n_t The number of time-steps in the time interval $[t_0, t_f]$.

n_u Number of states in the uncertainty state-space system.

n_x Number of cells in the spatial discretisation of $\partial\Omega$ in the horizontal direction.

n_y Number of cells in the spatial discretisation of $\partial\Omega$ in the vertical direction.

n_z Number of reduced order model states, or mode amplitudes.

p Internal pressure in a flow.

$P_{\hat{\vec{O}}_p}(x_i, y_j)$ The min-max normalised over $\partial\Omega$ probability of oil presence/drift for a cell (x_i, y_j), at a given time.

$q(\overset{u}{\vec{x}}, t, \vec{p})$ Oil uncertainty, as a function of the variances in particle position.

t The time, in seconds unless otherwise stated.

t_0 The start time of a simulation.

t_ϵ Sensor pathing optimisation end time.

t_ι Start time for a single sensor step form of the sensor pathing optimisation.

t_ϕ End time for a single sensor step form of the sensor pathing optimisation.

t_τ Sensor pathing optimisation start time.

t_f The end time of a simulation.

t_k The time at discrete time-step k.

u Horizontal component of velocity.

v Vertical component of velocity.

x, y, z The horizontal, vertical and depth position in Ω, increasing west-to-east, south-to-north and surface-to-sea-floor.

z_{crit} The depth at which the water depth discretisation switches from a fine to course mesh.

α_w Coefficient that attenuates the velocity of surface water resulting from wind induced surface shear.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This thesis describes a framework for the autonomous monitoring of contaminants in fluids. Though a broad topic, there is focus upon the hindcasting, estimation and prediction of an oil spill in a sea environment using mobile sensors. This is not a new problem to tackle, with oil spill models and surveillance tools in common industrial use since the reformation of the shipping industry following the Exxon Valdez oil spill of 1989, and long prior to that in more occasional use and research. Despite this longevity, it is still an active research field to which this thesis contributes. Before continuing, it is important to establish the process following a maritime incident.

Consider the first response crew to a vessel in distress. Unless the coastguard, emergency services or military are required to be involved, this is likely to be a private-sector team formed from concerned parties and their hired investigators; the insurers, charterers, the ship owner and cargo stakeholders. Their mission is to determine what has happened, what can be done about it and who is liable to pay the cost at the end. This holds true for anything from a minor ship-fire, to a collision, to a discharge of oil; accidental or otherwise. This is not an easy task due to the lack of knowledge around, taking an oil spill as an example, the spill location, leak time, leak amount and oil type. The past, current-time and future estimation of this spill fate and the response to it, are all extremely sensitive to these variables. It is an unfortunate truth that useful information is often obfuscated by a potentially liable party. This could be a sleep-deprived crewman who accidentally discharged the wrong hold tank into the sea, sending tonnes of oil into the ocean instead of sea-water, or the ship-owner who failed to see his vessel properly maintained: Getting to an accurate description of the incident can take time. Unless this is a

particularly severe incident, it is unlikely any expensive resources (aircraft, satellites etc) will be assigned until deemed absolutely necessary and this holds true for a government investigation also. Hence, the true extent of an oil spill is unlikely to be determined until much later in the time-line. Several days later is common, but then so is months or years. However, the arrival of increasingly low cost autonomous sensor platforms, in Unmanned-Airborne-Vehicle (UAV) or Unmanned-Submersible/Seaborne-Vehicle (USV) configuration, could give the first responders a method to rapidly gather wide-ranging information on a contaminant leak, sea conditions and another view on the situation. These autonomous platforms, together with advances in modelling, data assimilation and control theory, have opened the door to new approaches.

Regrettably, the advances in these fields are not always complementary. The hydrodynamic and oil models, used for environment flow and spill estimation, are becoming ever more complex, computationally intensive and restricted to access. To briefly describe the complexity of state-of-the-art models, they combine stochastic random behaviour, with one, two, three or four way coupling between wind, wave, water and contaminant behaviour, with flow described as 4-dimensional partial differential-algebraic equations and wave motion stemming from an overlapping spectrum of frequency and amplitude information. A decade ago, heuristic and empirical relations between, for example, the presence of oil and the dampening of the wave spectrum, were utilised. The field is now moving towards a fully physically coupled system. This is simply infeasible to use in a real-time manner, within a controller.

In contrast to the above, within industry the fluid and oil models are becoming increasingly separated, with it common practice to use operational hydrodynamic models as external data providers to an entirely uncoupled oil model. An operational model is simply one that is always in use, likely upon a high performance computing cluster, that makes regular data available for access. However data access is unlikely to be free for the most useful of data sets. This separation has made it difficult for any measurements of parameters outside of either the hydrodynamic, or oil model, to be assimilated and effects included within the counterpart model. In industry, there is seldom an online feedback loop between environment and oil measurements, model use and measurement tasking.

In the data assimilation literature, new methods are enhancing the ability to use sparse measurements, leveraging parallel model runs, reduced order models and/or machine-learning ap-

proximations of model derivatives to speed up the process. However, once again, the ever increasing model complexity can result in a loss of dynamic fidelity when stepping from the full model to the reduced order model, or limit parallel runs. In oil spill monitoring, machine learning is a risky approach when an opposing expert in a court of law can point to the lack of physical roots and data fitting inherent in data-driven methods.

Meanwhile, in the control field for mobile sensors, a huge variety of methods exist, exploring anything from bio-mimicry, to abstract information theory, to traditional model-based approaches. Each method has merits, but often lacks in two areas: the underlying model or assumptions are too simple to capture the dynamics of oil in the ocean, or the method fails to consider correcting past/future error in both contaminant and environment states. For an oil spill, determining what has happened accurately can be just as important as establishing the current situation and the future prediction. A final area often lacking in prior work, is consideration of the sampling most useful to enhance the accuracy of the underlying model upon which all other response decisions are made, not simply more energy efficient sampling. The control field for oil spill response can be divided into three broad categories, strategic guidance, tactical guidance and local guidance. This would be assigning sensors to an area of operations or spill site, determining a measurement path for a sensor around a spill and the actuation required to sense along a path in the presence of disturbances. This work will avoid the low level control of a sensor platform, instead assuming there is already a controller present to manoeuvre the platform and make use of the equipped sensors.

In this work a practical outlook is maintained while creating an oil contaminant monitoring framework, focusing on the use of sensor equipped UAVs. The decades of hydrodynamic and oil spill modelling is distilled into a very fast, adequately accurate, combined ocean and oil model. This model is validated against a real-world spill, and then is examined to extract a definition of uncertainty in its oil spill predictions. In a novel optimisation, the uncertainty is minimised by a sensing plan. These sensor measurements are assimilated into the model with a carefully selected and adapted method from recent literature, that utilises the combined and speedy properties of the model to employ reduced order modelling only where needed, thus maintaining dynamic fidelity. The monitoring framework is demonstrated to improve upon the industry standard method in a forward estimation and prediction scenario, a late-arrival

hindsight analysis scenario and finally a source locating scenario.

The remainder of this chapter first presents further motivation for the research, with industrial statistics and failings of the current methods detailed in Section 1.1. The formal research aims are contained in Section 1.2 and the chapter concludes with a description of the following chapters and their contributions in Section 1.3.

1.1 Motivation

Maritime incidents are varied and dynamic situations where immediate observation provides valuable information for assessment and resource allocation. The maritime industry is growing by approximately 3% per annum (UNCTAD 2016) incorporating more ships and larger vessels every year. However, with companies' revenues fluctuating around their operating costs, the need to be competitive can lead to dangerous accidents. There are approximately 10 maritime incidents every month, with 85 ships lost in 2015 and a total of 1231 ships lost between 2006 and 2015 (Allianz 2015). In addition to ship wreckage and their contained fuel and fluids, 10000 shipping containers are lost per annum, each one a navigation hazard. There are also around 7 oil spills every 12 months, in 2016 there were 4 recorded spills and 1 major recorded spill, totalling around 6000 tonnes of oil (ITOPF 2017). Note that The International Tanker Owners Pollution Federation (ITOPF) has the acronym ITOPF. The largest recent spill, excluding the Deepwater Horizon spill, was the Sanchi oil spill of 2018, of 116'000 tonnes (ITOPF 2019). Maritime incidents can lead to expensive court cases, argued through convoluted law that is difficult to litigate without hard evidence of a party's innocence or guilt. Verdicts often result in millions of dollars in damage settlements, with the record being the $5.5 billion in damages from BP as a result of the Deepwater Horizon oil spill. Clean-up operations, accident monitoring and rescue attempts are often hindered by the information and resources available at the accident locale, with specialist equipment including observation aircraft not arriving until several days after the event. The lack of information is partially mitigated by remote sensing assets, reviewed in Fingas and Brown 2014 and 2018.

Current observation solutions include satellites, capable of delivering a detailed view of an entire oil spill, through a variety of sensor types with the most common being optical, microwave and radar wavelength based. Though new satellites and data processing techniques are increasing

the frequency and reliability of measurements, the delay from tasking a satellite to image delivery is still up to 4 hours with an overpass frequency of once-per-day common (Fingas and Brown 2018; Carlowicz 2010). In the Sanchi tanker incident of 2018, the COSMO-SkyMed satellite system (Fiorentino and Virelli 2016) first provided Synthetic Aperture Radar (SAR) data on January the 15th, 9 days after the vessel registered distress and roughly 18 hours after the vessel sank (E-geos 2018). Evening data on the 15th and data on the 16th gave false clear-sea readings due to spill and vessel drift outside of the observed location (E-geos 2018). Good data once every 12 hours continued on the 17th, with SAR Sentinel 1-S1 imaging acquired on the 20th (E-geos 2018). SAR satellites are unreliable in calm or rough seas (wind speeds less than 3 m/s or greater than 10 m/s) and environmental phenomena can produce false positives (Topouzelis and Singha 2016). SAR is incapable of measuring oil thickness and the complex interplay between oil thickness, viscosity and wave parameters results in further uncertainty in measurement results (Zhang et al. 2015). Evidently, the utility of Earth monitoring satellites to first responders could be improved, as they currently suffer due to their delay and some dependency on oil trajectory prediction, and direct observation assets such as aircraft are preferred (ITOPF 2014).

Not all maritime incidents are reported immediately, or at all, with clandestine releases of oil into the ocean all too common. Often, first notification of an oil spill or wreckage comes from a network of SAR satellites that monitor shipping lanes (ITOPF 2014). Possible incident sites must be verified by direct observation, usually meaning aerial observation. However, due to remoteness, flyovers are often conducted using a local aircraft with no specialist sensors or tools, crewed by a human observer (ITOPF 2014). In extreme locations aerial observations are hampered by a lack of runways, requiring the chartering of helicopter pad equipped vessels, but still restricting the use of specialist fixed-wing aircraft and delaying observation by days, if-not weeks (Laruelle 2011). Once arrived, the expense of aircraft limits their number and hence the availability of simultaneous viewpoints or constant coverage during pilot/refuel breaks. Furthermore, health and safety concerns for the crew can limit their night-time deployment and their flight route is often pre-determined before take-off, with changes at the discretion of safety and airspace concerns. Observation aircraft plan routes as ladder search patterns in the supposed direction of wreckage or oil migration, usually estimated with large scale measurements

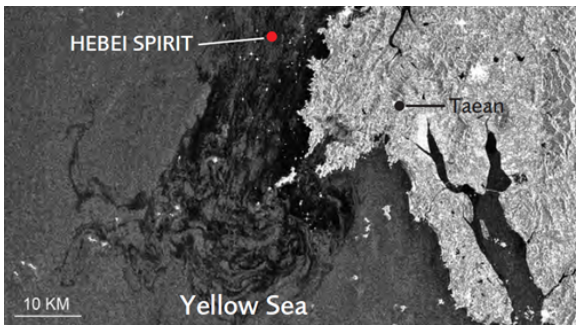


Figure 1.1: SAR image of the Hebei Spirit oil spill of the 7th December 2007. The image shows the oil spread on the 11th December 2007 (Laruelle 2009).

of current and wind and a very simple model of drift. Supporting tools, such as oil models, may not be available (due to a lack of data or resource allocation) in the crucial first few days of an incident. The complexity of existing models produces slow calculation times but useful data for response planning. However, despite their complexity and supposed accuracy, model predictions still have to be verified by observation before resource allocation in the industry (ITOPF 2014).

Failings of current solutions are visible in the Hebei Spirit spillage of 2007, pictured in Figure 1.1. The collision occured on the 7th December and was reported immediately. Coast guard vessels and non-specialist aircraft were on-scene after 4 hours and noted an oil leak; promptly beginning damage mitigation and vessel salvage efforts (The Hong Kong Special Administrative Region Marine Department 2007). Though rough weather hampered oil containment efforts, South-Korea's Ministry of Maritime Affairs and Fisheries predicted the oil wouldn't spread due to the cold weather (Bae Ji-sook 2007). By the 10th December, when the first ITOPF overflight and aerial observation using specialist equipment occured, the oil had spread 70 km along the coastline. Alerted to the spillages severity, the first SAR sateillite image, as seen in Figure 1.1, became available on the 11th December and showed the extent of the spread clearly (Laruelle 2009). Had specialist observation tools, with a supporting model, been available sooner the large scale of the spill and estimated drift could have directed resource allocation more efficiently from the outset.

1.2 Research Aims

The aim of this research is to produce an all-in-one framework for the autonomous monitoring of contaminants in fluids. Sensors must coordinate to be in the right place at the right time to sense contaminants, needing guidance from model-based predictions and data assimilation. This poses research challenges of:

- Choosing a monitoring strategy appropriate to maritime incidents, including the platform and sensor capability.
- Modelling contaminant movement using a sea state model, with prediction/analysis and source location estimation. The model must be computationally tractable to run in real time.
- Cooperative control of multiple systems to ensure an optimum coverage sensing strategy: Adaptive decision making considering sensor, platform and communication capabilities, considering their constraints in forming a sensing strategy. Navigational planning should include consideration of model or assimilation short-comings.
- A further challenge of how to sense and update the combined environment and oil model using new information and the incorporation of prior knowledge, such as spillage type, in the prediction. The assimilation method must also be fast enough to run in real time, and not require parallel runs or simplification of the model to a problematic degree.

Research outcomes include a simulation of sea contaminant scenarios and the sea surface and contaminant within, a method of assimilating measurements into these simulations, as well as a novel controller for guiding UAVs to gather these measurements that includes consideration of the inaccuracy in the model or assimilation methods.

1.3 Description of the thesis, contained work and contributions

This section describes the structure and contributions of the following thesis chapters, first in summary and then in detail. Chapter 2 presents the reader with background maritime

information and a review of prior, related research. Chapter 3 contains the overall monitoring framework structure, with a time-line of the algorithm when applied in the simulations of Chapter 7. Chapter 4 describes the modelling component of the monitoring framework. Chapter 5 details the sensor guidance stage of the monitoring framework and the optimisation which it entails. Chapter 7 contains an evaluation of the monitoring framework performance in various scenarios and Chapter 8 concludes the thesis.

Describing the chapters and their contributions in further detail:

- Chapter 2. First in this chapter, the use of UAVs and their sensor packages is examined in the maritime setting to provide the reader with valuable context for the capability of UAVs and remote sensing. This includes a general overview of UAV and sensor types and potential deployment purposes. A brief description of the optical appearances of oil on water is also given. The chapter moves on to a literature review of existing hydrodynamic and oil modelling, in the context of their suitability for a monitoring framework. The review of existing methods continues to sensor guidance, across general areas of moving to a fixed point, search and following, path planning and then the most closely related works involving contaminant tracking or clean-up. Then, the review briefly covers modern data assimilation methods employed in the hydrodynamic field, to offer context for the monitoring framework assimilation method. Finally, the chapter summarises the key insight from prior work.

- Chapter 3. This short chapter presents the monitoring framework as a whole, a brief overview of the utilised methods, and how the iterative procedure functions in practice. This structures the chapters that follow.

- Chapter 4 begins with an overall description of the environment and oil model, then details the spatio-temporal structure and states of the model. The Chapter moves on to the implementation of the 2D fluid flow solver, including the boundary conditions and the expansion of a surface velocity to a 2.5D description with a depth velocity profile. Further components of the fluid model are detailed, including wind induced phenomenon and the wave model. Next, the oil model component of the combined model is examined, with advection and diffusion, entrainment and buoyancy, thickness and mechanical spreading, deposition and refloating all described. The surrounding parameters of an oil model are

then looked at, that is, the number of particles needed, and extracting probabilities or loosely defined properties (like the mean spill location) from the model. The backwards implementation of the model is then noted. The chapter then presents a validation of the model against a real world spill and a comparison with the common industry model GNOME (General NOAA Operational Modeling Environment). Finally, the chapter extracts the definition of uncertainty and a companion description of sensing, for use in sensor pathing, then represents uncertainty as a state-space system. The key contributions of this chapter is first, the development and validation of a bespoke model of wind, ocean and oil dynamics for real-time monitoring, and secondly a PDE description of the uncertainty of an oil particle distribution.

- Chapter 5 describes the sensor guidance component of the monitoring framework, and the presentation and solving of a weighted, constrained uncertainty minimisation optimisation. The optimisation is introduced, then the weighting terms and constraint Lagrangian multiplier described. The solution method, both the adjoint method based gradient determination and gradient descent optimisation is then detailed, with example solutions under static and mobile sensors then presented. The optimisation formulation and solution method is a key contribution of this thesis, as it presents a multi-scale approach suitable for guiding both oil measurements and fluid measurements in support of oil monitoring.

- Chapter 5. This chapter sets out the data assimilation method for the monitoring framework, first giving an overview, then recounting the formation of the reduced order model from the full order model state trajectory. The reduced order model is then used with an analysis capable state estimation method capable of assimilating a trajectory of measurements and external values. Finally, the assimilation of measurements and external values to form a complete full-order state trajectory estimation concludes the chapter. The contributions in Chapter 5 are extensions of state estimation and reduced order modelling techniques to develop an ensemble and tangent-linear model free data assimilation method. A further key idea is to formulate the error of the data assimilation method and utilise the error to inform sensor placement.

- Chapter 7 evaluates the complete monitoring framework against industry standard methods of pathing and data-assimilation, across a forward estimation and prediction scenario, a late-arrival hindsight analysis scenario and finally a source locating scenario. The monitoring framework is demonstrated to improve upon the industry standard method in both spill estimation accuracy and sensor utilisation. This chapter demonstrates the monitoring framework reduces estimation error (using a combined measure of spill position and shape error) by up to 80%, when compared to a traditional ladder path sensor guidance method and/or a simple data assimilation method.

- Chapter 8 concludes the thesis, by reiterating the principal research results and expanding upon the contributions, then suggests avenues for future research.

To summarise the novelty of the thesis, in the context of real-time oil spill monitoring with mobile sensors and computing hardware: There are a number of incremental improvements in each of the main focus areas, which are wind/wave/oil modelling, oil uncertainty description, sensor guidance optimisation and solution, reduced order modelling and finally data assimilation. The work is detailed in the relevant chapters with brief comments on their contributions, though the thesis contributions are elaborated upon in Chapter 8. However, the main contribution is the linking of each component to form the monitoring framework as a whole. This is an all-in-one monitoring framework that tackles the difficult problems of oil spill source determination, hind-casting, now-casting, prediction, through the guidance and utilisation of mobile sensors.

1.4 Notation

Regarding notation: Where a function is presented with explicit arguments within an equation, arguments are present to emphasise to the reader the dependency of said function upon particular variables. This dependency is important when considering the sensitivity of a system with respect to the optimisation variable. In the interest of notational brevity, not all arguments will be presented within an equation, but all functions are formally defined in the main text. For example, given the scalars $x, y \in \mathbb{R}$ and a function $f(x, y)$ where $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, the function $f(x, y)$ may be described as $f(x)$ within an equation. Consider the equation, $\min_x f(x)$, where only dependency upon x is explicitly noted. Furthermore, integer intervals are denoted

by double brackets, e.g, $[\![a, b]\!] = \{a, a+1, ..., b-1, b\}$. Additionally, given a matrix $\boldsymbol{M}$ of size m by n, a set of row indices $\vec{\mathscr{I}}_r \subseteq [\![1, m]\!]$ and a set of column indices $\vec{\mathscr{I}}_c \subseteq [\![1, n]\!]$, then the notation $\boldsymbol{M}(\!|\vec{\mathscr{I}}_r, \vec{\mathscr{I}}_c|\!)$ describes a sub-matrix $\boldsymbol{M}(\!|\vec{\mathscr{I}}_r, \vec{\mathscr{I}}_c|\!) = \{\boldsymbol{M}(\!|i, k|\!) : i \in \vec{\mathscr{I}}_r \wedge k \in \vec{\mathscr{I}}_c\}$. The set of positive real numbers including 0 is defined by $\mathbb{R}_+ \subset \mathbb{R}$. Throughout this thesis, $(\cdot)^T$ is the transpose, $(\cdot)^*$ is the conjugate transpose or Hermitian, $(\cdot) \circ (\cdot)$ is the Hadamard product or elementwise product and $(\cdot)^{\circ k}$ is the Hadamard exponential to power k. Further notation includes, $\hat{\cdot}$ as an estimate and $\bar{\cdot}$ as a centered value or mean value with clarification in the text. Vectors are denoted $\vec{\cdot}$, matrices in bold capitals, sets or trajectories in calligraphic. Any exception to this notation is explicitly stated. The vertical concatenation of vectors, $\vec{c} = [\vec{a}^T, \vec{b}^T]^T$ is denoted by $\vec{c} = [\vec{a}; \vec{b}]$.

Consider the vector $\vec{c} \in \mathbb{R}^{n_c}$ of $n_c \in \mathbb{N}$ elements, with the ith element $\vec{c}_i \in \mathbb{R}$. If the ith element is then subject to bounds $\vec{c}_i \in [l, u]$ for $l \in \mathbb{R}$ and $u \in \mathbb{R}$, then an under-bar notation, such as $\underline{\vec{c}}$ explicitly denotes that $\underline{\vec{c}}$ is formed of bounded values, while $\vec{c}$ is not. This is relevant when describing bounded and unbounded state vectors.

Chapter 2

Background and Related Work

This chapter first presents background maritime information, commenting on the common scenarios and challenges facing autonomous monitoring in a maritime deployment. The sensor requirements and capabilities are also noted, together with the visual appearance of oil spills of varying thickness. The chapter then delivers a review of current oil spill and hydrodynamic modelling methods, providing context for discussion of sensor guidance methods in a further review and finally a review of the assimilation of sensor measurements into an oil spill and hydrodynamic model.

2.1 Maritime information

The use of autonomous vehicles, airborne, seaborne or land based is fast becoming widespread in the civilian, commercial and military markets. Though largely successful in military and government use, their deployments alongside civilian operators have been hampered by legislation, safety concerns and the need for an operator to maintain constant control authority; both to ensure danger avoidance and to adapt the automaton's planning to environmental changes. This is of particular prevalence for small Unmanned Aerial Vehicles (UAVs) and Submersibles (USVs), where the real time, 2-way exchange of large amounts of data becomes infeasible due to their mission range and the capabilities of their communication suite. Effort must be made to improve the true autonomy and self-governance of UAVs and USVs in dynamic missions, both in high-level decision making and low-level route planning (Zeigler 1990). The current trend in deployed autonomous systems is to utilise low numbers of high cost, high sensing fi-

delity and high performance systems that maintain 2-way communication and control to enable mission flexibility and operational safety (Moreland et al. 2015). While capable of acquiring excellent data the low number of viewpoints does not allow simultaneous coverage of a large area and cooperative search methods are excluded. While industry acknowledges the potential of autonomous systems, uptake is slow until the technology and surrounding legislation matures (Foxwell 2017).

Developing autonomous systems faces research challenges under headings of energy limitations, environmental and operational hazards, information processing and human-system interaction. These problems are exacerbated in a distant environment (Kitano et al. 1999). A flexible, cooperative and predictive package containing navigation, sensing and control algorithms could enable far reaching missions of multiple autonomous vehicles, operating efficiently in a dynamic scenario with minimal input from a human controller beyond what is currently capable in the industry.

UAVs are particularly suited to use in the maritime domain, offering a range of benefits across all their size classifications (Kaymal 2016). UAVs offer a high area coverage in a short amount of time, due to their high speed (relative to ships) consistent during their deployment. Depending on their size, UAVs can offer extremely long endurance flights, without personnel based limitations. Even smaller UAVs can maintain coverage for long periods if multiple systems are deployed with overlapping coverage during battery or fuel replenishment. The low price per unit and small size compared to manned aircraft or surface vessels facilitates the use of multiple UAVs simultaneously, allowing for simultaneous view points across a wide area and more efficient search techniques (Cevik et al. 2013).

UAV design for oil spill monitoring is determined by a number of factors; with the information required perhaps the most critical: The absolute presence of oil measured as fast as possible across a large area, or the thickness determined in a small area, or the oil type ascertained at source. All would produce a different ideal design, coupled with other environmental characteristics such as legislation or shared airspace. There exists a number of common maritime incident scenarios where a UAV could provide valuable insight or improve upon the current solutions employed by industry, satellites or manned flights. Each has a corresponding ideal UAV design, portrayed in table 2.1.

UAV design considerations for maritime incident scenarios		
Incident description	Design parameter	Requirement
Close proximity survey and inspection of spill source	Flight altitude	Very low (50 m max)
	Payload	7 kg max
	Endurance	1 hour max
	Platform type	Multicopter for stable, close up sensing and access to unique locations
	Sensor	High resolution optical sufficient
	Communication	Short range, high bandwidth for data fidelity
Short distance survey and source location and inspection	Flight altitude	Low (100 m max)
	Payload	25 kg max
	Endurance	4 hour max
	Platform type	Fixed wing for endurance, range and payload
	Sensor	Optical/IR/Lidar
	Communication	Mid range, mid bandwidth for data range. High fidelity imagery can be burst transmitted at the expense of other data, a spills location, thickness and source is sufficient
Long distance survey, source location and inspection, and contaminant assessment.	Flight altitude	Legislation limit (400m max)
	Payload	100 kg max
	Endurance	30 hour max
	Platform type	Fixed wing for endurance, range and payload
	Sensor	Multisensor/Deployable
	Communication	Long range, low bandwidth for data range. High fidelity imagery can be stored for review, a spills location, thickness and source is sufficient

Table 2.1: UAV design considerations for maritime incident scenarios, adapted from Gómez and Green 2017.

Minimum spatial resolution requirements (m)		
Task	Large Spill	Small Spill
Detect oil on water	6	2
Map oil on water	10	2
Map oil on land/shore	1	0.5
Tactical water clean-up	1	2
Tactical support land/shore	1	0.5
Thickness/volume	1	0.5
Legal and prosecution	3	1
General documentation	3	1
Long-range surveillance	10	2

Table 2.2: Requirements of oil spill detection.

Typical airborne sensor characteristics			
Sensor type	Sensor	Spatial resolution range (m)	Swath Width (km)
Radar	SLAR	10-50	10-40
Radar	SAR	1-10	10-40
Optical	Video Camera	< 1 & Alt. Dep.	Alt. Dep.
Optical	Still Camera	< 0.1 & Alt. Dep.	Alt. Dep.
Optical	Typical Ultraviolet/Infrared Scanner	< 1 & Alt. Dep.	Alt. Dep.

Table 2.3: Typical airborne sensor characteristics. In the table, Synthetic Aperture Radar (SAR) and Side-Looking Airborne Radar (SLAR) use their acronyms and "Alt. Dep." is altitude dependent.

Contaminant detection

The expected contaminant and mission profile determine the type and quality of sensors needed aboard the UAV platform. Prior research offers sensor resolution requirements in table 2.2 and typical resolution of existing sensors in table 2.3 (Jha, Levy, and Gao 2008).

Payload weight and size constraints apply, but a plethora of possible sensors remain. Despite this, simple still or video photography is the most common form of contaminant or oil spill detection (Fingas and Brown 1998), using the empirical observation rules in table 2.4. Note the orders of magnitude volume difference between oil appearances: although sheen may cover large areas of the environment, it is a negligible portion of the total oil volume. Hence observation should strive to focus on dark patches and be able to identify between sheen and thicker patches of oil. Supplementing optical observation, airborne remote sensors can offer oil detection, or information on the parameters of oil, in conditions where the visible spectrum is unsuitable, such as night time or rough sea conditions. Sensors can be categorised as passive, or active;

Optical appearance of oil			
Oil type	Appearance	Approximate thickness (mm)	Approximate volume (m^3/km^2)
Oil sheen	Silver	> 0.0001	0.1
Oil sheen	Iridescent (rainbow)	> 0.0003	0.3
Crude or fuel oil	Brown to black	> 0.1	100
Water/oil emulsions	Brown	> 1	1000

Table 2.4: A tabulation of the thickness, appearance and volume of floating oil when observed from the air. Adapted from ITOPF 2011a.

their capabilities are tabulated in tables 2.5 and 2.6 respectively. The correct selection, or further development of sensors will allow for accurate detection of surrounding fluid velocities and contaminant properties. It is worth noting that sensor capabilities are improving all the time and the sensor data above may become obsolete. Recently launched satellites may offer a much greater resolution in the visual and SAR spectra, but satellites are still limited by weather conditions (cloud cover) and the narrow band of wind conditions and false positives inherent in SAR oil spill detection (Topouzelis and Singha 2016). However, even a perfect sensor is useless if it is not employed in the correct place at the correct time, which is the purpose of guidance algorithms.

Guidance Algorithms

The maritime industry currently employs surveillance aircraft to map contaminants if possible and the resources are available. Designated aircraft range from specialised, multi-engined observation craft with a trained crew and dedicated sensors to light aircraft and the human eye. Their guidance is simplistic, usually following a pre-planned ladder path over a region identified by a spill trajectory model, or spiralling inwards/outwards to an estimated source location (ITOPF 2011a). While ensuring good coverage, these are time consuming and inefficient. A further consideration to be included in the decision making process would be the robust communication between relevant agents. Control might be discretised, but communication and information sharing is likely still essential to ensure efficient area and target coverage. Hence constraints should be present upon sensor positions, to ensure they are within communication range.

A summary of passive sensors suited to sensing of oil and other contaminants			
Type	Typical task	Advantages	Weaknesses
Passive Visible (wavelength $0.38 - 0.76\mu$m)	Spill detection Approximate thickness detection Approximate volume estimation Optical flow estimation	Visual data Inexpensive Light weight	Limited to good visibility conditions, e.g, affected by darkness, clouds, haze or smoke
Short-wave infrared (wavelength $0.9 - 1.7\mu$m)	Spill detection Relative thickness detection Approximate volume estimation Capable of differentiating between crude oil and weathered emulsions	Suitable for low light conditions Low power consumption Light weight and small size	Expensive to produce and operate Not visible to human eye, information must be provided as spectral maps etc
Thermal infrared (wavelength $8 - 14\mu$m)	Night-time spill detection Approximate thickness detection Approximate volume estimation	Suitable for low light and night-time conditions Vision through clouds, haze or smoke Low power consumption Light weight and small size	Reference data for comparison is needed Not visible to human eye, information must be provided as spectral maps etc
Live video	Monitoring situation Approximate thickness detection Approximate volume estimation	Forward looking video would allow remote piloting if needed Possibly 3D information, produces measurement of wave data	Reduntant information a lot of the time Lower spatial resolution than still imagery

Table 2.5: A tabulation of a selection of passive sensors suited to contaminant detection.

A summary of active sensors suited to sensing of oil and other contaminants			
Type	Typical task	Advantages	Weaknesses
Active Lidar	Spill detection Absolute thickness detection Oil identification (Raimondi et al. 2017) 3D position and velocity measurements	High precision	Limited to good visibility conditions, e.g, affected by clouds, haze or smoke High power consumption Extremely accurate position data required for measurements Lack of suitable commercial sensors Miniaturisation difficulties
Radar	Large area detection of contaminants and oil spills	Day and night conditions Unaffected by smoke, haze or clouds.	Power consumption Unreliable in low or high wind conditions Not visible to human eye, information must be provided as spectral maps etc

Table 2.6: A tabulation of a selection of active sensors suited to contaminant detection.

2.2 Modelling

This section presents an overview of oil spill modelling and the hydrodynamic models that oil spill models depend upon. A broad overview of spill models is followed by a more in-depth examination of the modelling approaches for the physical processes important to oil trajectory extrapolation. The section continues to similarly examine the modelling approaches for the environmental physical processes: The description of the wind and ocean dynamics that govern the movement of an oil spill.

2.2.1 Oil spill modelling

Spill modelling for a particular incident typically ensues with a pre-validated oil spill model. There are several available, from commercial and proprietary models to academic and open-source models. Each model is the outcome of a significant investment in representing the physics of an oil spill in mathematical form, with some models being specific to a particular region and the local dynamics in that area. These models produce an estimate trajectory of the oil through a spatio-temporal domain and form the basis for stochastic measures and decision making. This could be, determining the likelihood of oil reaching a certain region, or a measure of the likelihood of a significant volume of oil remaining (Nelson and Grubesic 2020).

A recent state of the art review of oil spill modelling (Spaulding 2017) covers OSCAR (Reed et al. 2000), SIMAP/OILMAP (French McCay et al. 2016), GNOME/ADIOS (Lehr et al. 2002), though other notables in the field include the model, MEDSLIK (De Dominicis et al. 2013a) and BLOSOM (Duran et al. 2018). There is also a wealth of commercial oil models, with propriety methods for which little information is available.

The review by Spaulding 2017 affirms modern oil spill models are complex amalgamations of Lagrangian (particle based) transport processes and varied algorithm types (stochastic and deterministic) of other processes, such as entrainment in the water column, or evaporation. There are some exceptions that use an Eulerian approach (Taylor et al. 2003), but these are more limited in scope as supporting algorithms (such as entrainment) are Lagrangian based (Wang and Shen 2010), providing solutions per particle. State of the art 3D models aim to provide the most accurate estimations possible of oil position/properties, both surface and sub-surface, at the expense of computational speed, over an extended period of time (weeks

to months of prediction) and hence include weathering effects. Despite their complexity and supposed accuracy, their results still have to be verified in the field; the industry will not allocate resources based on modelling alone (ITOPF 2014).

The modelling of the oil spillage itself can be sectioned into principal physical processes, in order of importance with increasing time moving through a surface spill scenario: mechanical spreading, advection by wind/wave/current, turbulent diffusion, entrainment (including buoyancy and droplet size) and evaporation. Dissolution, emulsification, biological effects and photo-degradation are other phenomona, but play a much lesser role in the first 24 to 72 hours of an oil spill (Proctor, Flather, and Elliott 1994).

Mechanical spreading, without external forces, is the process of oil spreading over a surface, to form a flat circle where internal hydraulic pressure and boundary surface tension balance. Spreading is usually modelled with empirical formulae, producing an estimated radius at a given time for a thick slick (Fay 1971). Field observations or past data provide a spreading rate coefficient and spreading ceases when the oil reaches a minimum thickness. These methods have significant shortcomings, being only valid in calm conditions and ignoring the effects of wind and current on shape (Hoult 1972). Later methods expanded upon Fay's work, to produce elliptical distributions (Lehr et al. 1984). Together with simple advection-diffusion equations, these form the common simple models used by guidance algorithms under the presence of uniform flow, for example within Zhang and Pei 2015.

Advection is a collective term for the mass-transport of oil particles by the wind and hydrodynamic model. Across reviewed models it is described by a vector summation of water current velocity (scaled by an empirical coefficient), the wind velocity (scaled by an empirical coefficient and rotated by an empirical drift angle) and a force or velocity from the wave model, wave radiation stress or Stokes drift (Spaulding 2017). Coefficients vary with literature, from 1 to 1.1 for current velocities in (Lonin 1999) and (Wang, Shen, and Zheng 2005) respectively, and 0.01 to 0.06 for wind velocities and some geographical variation in drift angle formulation, from 0 to 45 degrees (Li, Zhu, and Wang 2013). Note however, that the drift angle should be omitted in cases where the underlying environment data already incorporates Ekman currents, the off-wind-axis flow that results from the balance between coriolis forces and wind shear.

Turbulent diffusion addresses the turbulent processes that move oil particles on a smaller scale

than the grid of the hydrodynamic model. Almost exclusively, a Markov chain, or random walk technique is used with coefficients either described by the user or calculated from the hydrodynamic model. These coefficients vary by several orders of magnitude in the literature and play an important role in oil spill simulation (Spaulding 2017), for example, Elliott 1986 uses 0.7 m^2/s while Al-Rabeh, Cekirge, and Gunay 1989 uses 10 m^2/s as constant parameters. Smagorinsky's non-linear diffusion equation is present in some models to calculate the diffusion coefficient. While dependent upon flow derivatives, it's also strongly sensitive to a dimensionless parameter set by the user, for which literature suggests values ranging from 0.03 (Baldauf and Zängl 2012) to 0.2 (Wang, Shen, and Zheng 2005). Identified in Hunter, Craig, and Phillips 1993, and then further in recent work (Nordam et al. 2019b), there are failings in the commonly used diffusion coefficient formulae which can both under or over-estimate the true diffusion in both horizontal and vertical directions. The corrections developed in the literature have been applied here.

Wave induced motion calculation varies in literature. In early work, it might be omitted completely (Lardner and Gunay 2000), or simulated by Stokes drift calculations only using either wave parameters or empirical estimates from wind speed; this forms the majority of horizontal transport and is often taken as sufficient (Boufadel et al. 2007). Mellor's work advocates the inclusion of wave radiation stress (Mellor 2003) and recent work attempts to join Stokes drift and wave radiation stress to describe Langmuir cells, discussed in Galt and Overstreet 2011, that create the surface windrows (very thin lines of thick oil), though this was yet to be implemented into a model as of Spaulding's review (Spaulding 2017).

Subsurface entrainment, or subsumption of the oil into the water column, has several approaches to modelling in the literature, depending on the available data from the hydrodynamic model or measurements. Modern 3D hydrodynamic simulations can be coupled with a vertical diffusion process alone, while early work required empirical probability functions of particle entrainment, using wave energy and oil type, to ensure oil entered the sub-surface, as well as a vertical diffusion process (Li, Zhu, and Wang 2013). The work of Li, Zhu, and Wang 2013 compares 3 methods of vertical diffusion coefficient calculation, an empirical scheme, an internal hydrodynamic model solver and solving a Langevin equation. Results were inconclusive and failed to identify a most-accurate algorithm, though large differences in estimations were certainly

noticed, with the empirical method being the simplest to calculate. Buoyancy methods again vary, noted in Nordam et al. 2019a, where Nordam again identifies inconsistencies and correction modifications, implemented in this thesis.

Oil weathering or oil aging is a term for changes in oil properties over time, that includes evaporation, dissolution, emulsification, biological effects and photo-degradation. Weathering was determined to be non-critical in this research, due to the short horizon of the simulation and ability to account for changes in oil parameters through sensing. However, under some conditions, rapid evaporation can remove up to 40% of the oil mass within the first 2 days, if not a few hours for a lighter hydrocarbon such as Kerosene (ITOPF 2011c). Hence, estimations of weathering and particularly evaporation could be useful. A recent review of evaporation methods, from Mackay's early work (Mackay and Matsugu 1973) onwards, and investigation into oil evaporation (Fingas 2015), showed that oil evaporation is not strictly boundary-layer regulated, hence the dominant factors are time and temperature and so a simplistic evaporation description would suffice. Fingas 2012 offers several logarithmic, empirical equations improving on the over and under estimation of Mackay's oil component based work. That said, Spaulding 2017 argues that Fingas' methods has data requirement flaws and is verified with too much reliance on laboratory data. To account for weathering, the oil model here-in is capable of interfacing with the ADIOS weathering system within the GNOME oil model (NOAA 2012), by the transferance of particle and environment data.

In most cases, though limited to heavy, crude and other persistent oils, the majority of oil volume is contained on the surface, in dark slicks (ITOPF 2011a), with only 10% in the water column after 24 hours (Proctor, Flather, and Elliott 1994). When subsumed underwater temporarily, depths rarely exceed 10 m even in high wind conditions (Li, Zhu, and Wang 2013). This suggests a 2D current simulation, with empirical variation in depth, a 2D wind simulation and a surface wave model, would be sufficient for surface input data into a short-term oil model. The model would be intended for sensor guidance, not comprehensive spill simulation. Sensitivity studies of a similar model (De Dominicis et al. 2013b) demonstrate that a calibrated model retains predictive accuracy for approximately 1-2.5 days, with the forecast accuracy largely dependent upon the input ocean currents.

Backwards in time models have been attempted in literature, where in general, oil at a receptor

node, or final time position, is backtracked through a velocity grid produced by a forwards simulation of fluid. Early work relied on time reversed stepping of particles through time-invariant maps of mean current and wind fields to determine probability maps of oil starting at that position reaching a receptor node target (Galt and Payton 2005). Batchelder 2006 used time-varying velocity fields stored from a forward run of an ocean model to implement a forward and backward in time trajectory model of particles, that included advection and vertical diffusion (Batchelder 2006). Batchelder used a negative time step for advection and discusses the difficulty and utility of inverse diffusion, before including diffusion as a random process in his backwards in time simulations, to demonstrate its utility in describing an area of possible sources. Batchelder 2006 notes that vertical diffusion creates a particle dispersion with no indication of particles' initial depth after only 0.2 days of forward simulation and utilises a further forward simulation of his estimated source and the closeness of the resulting mean particle position to the original receptor node as a performance indicator.

Similar work, such as a two-way particle tracking model (Isobe et al. 2009), included horizontal diffusion as a random walk in both the forward and backward simulation. Multiple particles were reversed from a receptor node and each of their positions at the supposed release time was considered a source. These sources were then tested via forward simulation and a statistical measure of their final particle layout used to discard unlikely sources (Isobe et al. 2009). The above approaches were limited to a singular receptor node in their backwards time simulation. Perhaps the distribution of particles across multiple nodes could be used to infer something of the random process. A more recent work looks at utilising multiple receptor nodes, but only to generate probability maps of oil start positions reaching certain nodes (Ciappa Achille; Costabile 2014). Yu et al. 2017 describes a method of parameter estimation for wind and current coefficients, as well as random step size, given a large set of drift data. These parameters are then used in another reverse advection and turbulent diffusion by random walk (Yu et al. 2017). Inputs to the models, including geographical, wind and water current data, all must come from exterior hydrodynamic models, that also vary in approach.

2.2.2 Hydrodynamic modelling

Environmental models provide the oil spill model with wind, wave and current data. Modern 3-dimensional models commonly use a harmonic water-level tide model for boundary in-flows and out-flows and base their physical processes on the work of Mellor (Mellor 2003): 3D Navier Stokes, radiation stress from linear surface waves and a Smagorinsky eddy parametization, but with differing discrete solution methods such as an unstructured mesh (Wang and Shen 2010). Continuing work enabled coupling the wave model with an ocean model, and modification to incorporate depth induced wave breaking and wave-current interaction (Mellor, Donelan, and Oey 2008). Wave models are still external to the ocean model in most cases (Spaulding 2017), with one notable exception being Mellor's continuing research. This is a joining of the Stevens Institute of Technology Estuarine and Coastal Ocean Model (sECOM) and Mellor-Donelan-Oey (MDO) wave model (Marsooli et al. 2017). Some work omits Ekman currents completely (slow forming horizontal net water currents due to the force balance between the Coriolis effect and wind shear), others prefer to account for them (instantaneously forming) in their oil drift angle formulation (De Dominicis et al. 2013a), while others include them in their 3D hydrodynamic model by including a Coriolis force term in their Navier Stokes equations (Marsooli et al. 2017). Due to the constraints on computation, communication and time, the current 3D hydrodynamic models are unsuitable for UAV application: A state of the art model takes 74 hours to solve a 9 day simulation across 66000 nodes (the most useful measure of area), or approximately 400 km x 300 km, on an 8 CPU OpenMP computer (Marsooli 2017).

The separation of Ocean modelling to Oil modelling does have advantages, allowing for differing hydrodynamic approaches to be used and the appropriation of data from any source, be it small scale Boussinesq models (Lonin 1999), large scale circulation models (Marsooli et al. 2017) or broad-scale measurements: high-frequency radar, synthetic aperture radar (SAR), wave buoys or other data sources. However, there are disadvantages: If the models are not integrated, or run at the same time-steps, large data-sets must be produced and stored by the hydrodynamic model for use by the oil model, which may need to interpolate the data. Also, there can be no two-way coupling between oil and hydrodynamics; the dampening affect of oil on surface waves (integral to SAR measurement) (Zhang et al. 2015) cannot be included if the hydrodynamics are pre-calculated. Furthermore, certain parameters may only need to be calculated where oil

is likely to be. Wave spectra for example, could be calculated only where required.

The chosen approach here is a computationally tractable 2D hydrodynamic model to resolve input ocean and wind flow (if available from operational models) around local bathymetry and geography features too small to be included in the input data. Flow simulation in 2D has been conducted by the computational fluid dynamics (CFD) community as structured meshes, unstructured meshes (Chen, Liu, and Beardsley 2003), finite difference, finite element, finite volume and spectral methods. Assumptions that the sea-surface is inviscid, incompressible and irrotational are common and acceptable, though flow around geographical boundaries may (but not necessarily) invalidate this paradigm (Hover and Chin 2009). Since the priority here is large-scale fast simulation, focused on only the key physical processes relevant to the spatio-temporal evolution of oil spills, literature from the computer science industry that attempts to simulate realistic flow in real-time or faster is of particular interest. Classical CFD approaches that rely upon steady flows are disregarded, as tidal boundaries discount time-averaged methods. The methods of discretising the simplified, incompressible and irrotational Navier Stokes PDE vary from simple finite volume methods (Stam 2003) to even simpler, viscosity discarding Euler simulations (Braley and Sandu 2009). Because of its computational speed, a structured mesh (generated from images or bathymetry data), finite difference, projection based pressure correction solver will be implemented. The large scale of oil spills implies large mesh scales and long time-steps, while still offering a stable simulation using explicit methods.

The 2D ocean flow is then extrapolated to a complete vertical velocity profile to the sea-bed, using tidal current, Ekman current estimations, Stokes drift and wind induced surface shear. Additionally, a complete wave spectra is calculated where oil particles are present and environment conditions are contained within each grid cell, though use of spatio-temporally varying external data is also supported if available. The vertical velocity profile is important to estimate the further dispersal of oil resulting from its subsumption and resurfacing within water, without utilising expensive 3D flow simulation or large 3D external data-sets that are often unavailable for the local region.

2.2.3 Summary and further considerations for modelling

Concisely summarising the above reviews of oil and environment modelling: Both fields have a vast array of methods, spanning from simple empirical formulae to stochastic PDE solving in 3-dimensions. Environment modelling has clearer trade-offs between accuracy and computation speed, for example, the difference between a coupled 3D wave, wind and ocean flow model and a 2D uncoupled simulation of the water surface flow with an empirical wave model. However, in the oil modelling field every modelling approach has variation in coefficients, formulation and solution methods. There is little consensus on the most accurate approximation, as every oil spill is a unique example with difficult sensing challenges, limiting the comparison of methods to a known result (Spaulding 2017).

Here, the oil and environment models form part of a complete online monitoring framework. Hence, there are further considerations outside of accuracy and computation speed. The models must be capable of assimilating measured data, including correcting externally provided fluid flow data. Current stand-alone oil models are incapable of doing this without access to the external hydrodynamic models upon which they rely. Additionally, the model must provide stochastic measures of probability, as is common, but also of other information theoretic measures: Particle distribution variance and entropy for example. These further measures are valuable when determining sensor location.

2.3 Sensor Guidance and Optimisation

Efficient sensor guidance is critical to monitoring contaminants, ensuring sensors and payloads are in the optimal place and time. Sensor guidance can be divided into a hierarchy of strategic guidance, tactical guidance and local guidance. As described in Chapter 1, this work will focus upon the strategic and tactical level guidance; Assigning sensors to spills appropriately and determining a measurement path for a sensor once there, to map, monitor, predict and hind-cast the spill. The local guidance is assumed adequate, whereby the sensor can follow the planned path and gather measurements appropriately.

2.3.1 Sensor placement literature

Before focusing on sensor placement or guidance for contaminant monitoring, some background on the sensor placement research field. It is a vast and fast-moving body of work, with distinctly different approaches depending on the purpose of the sensing. The field expands again if the control of sensors, via Model-Predictive-Control or similar (Vali et al. 2019), is considered also. The method of sensor control or sensor placement varies with the intended use to such an extent that it is difficult to direct the reader to a single review paper, but perhaps Yi and Li 2012, Hinson 2014, Liu, Yan, and Guedes Soares 2018 and Wang, Li, and Chen 2020 offer an overview. To provide a brief summary, in Wang, Li, and Chen 2020, it splits the NP-hard problem of sensor placement under the broad approaches of decomposition based methods, optimisation methods, greedy methods, heuristic methods and machine learning methods. It is also relevant to briefly mention the metrics of sensor performance. Sensors are placed or pathed to fulfil a variety of conditions in literature, these include but are not limited to; 1) maximising the Fisher information matrix determinant, 2) minimising the estimation covariance matrix determinant, 3) maximising the observability Gramian, 4) minimising the highest variance in the estimation covariance matrix and 5) minimising the average variance in the estimation covariance matrix. So far, 1), 2) and 3) are all analogous and 1-3), 4) and 5) form conditions for D-optimal, E-optimal and A-optimal sensor placement respectively. There are further information theoretic metrics used, 6) maximising the information quality gathered and 7) minimising the information entropy. The reader is referred to Yi and Li 2012, Hinson 2014, Hollinger and Sukhatme 2014 and Leyder et al. 2018 for further details on these metrics, though there is discussion of which metrics intuitively suit this project in Chapter 5.

In this work, sensors are mobile, there is a short (a minute at most) time-constraint on calculation time, the environment and oil model is stochastic, non-linear and has changing numbers of states in its full-description due to the Lagrangian particle description of oil. Furthermore there are fundamental choices to make in a sensing strategy in an uncertain environment: exploitation, or exploration, and in this case whether to measure oil or environment. For example, consider the application of decomposition methods to the state trajectory of a combined ocean and oil model. An initial decomposition, whether Proper-Orthogonal-Decomposition (Daescu D. N., Navon 2006), Dynamic-Mode-Decomposition (Iungo, Abkar, and Port 2015), Empirical-

Orthogonal-Functions (Rozier et al. 2007) or similar (Galerkin etc), provides a reduced order description of the system in a set of spatially distributed basis functions, or modes. Assuming a truncated decomposition, to a handful of modes, the mode shapes would describe the principle flow structures of the ocean model and the major concentrations of oil. Additionally an approximation of the time dynamics could be constructed. Assuming a cyclic tidal ocean flow, the ocean flow time dynamics could be well described to an infinite time horizon, but oil dynamics are not cyclical and the decomposition model would fail when predicting oil distribution to a future time. However, the ocean flow and oil concentration mode shapes are still potentially useful. Two decomposition approaches to sensor placement for measuring ocean flow in Yildirim, Chryssostomidis, and Karniadakis 2009 and Clark, Kutz, and Brunton 2020 place sensors at mode extrema (highest fluid velocity) and the pivot points in a QR-factorisation of the decomposition modes respectively. In both cases, this is a sensing strategy focusing on accurate estimation of the modal amplitudes, not determination of the mode shapes in an unknown environment. Application of such sensing strategies to the decomposition of the ocean and oil model would place sensors at the supposed peak fluid flow velocities and oil concentrations for the method in Yildirim, Chryssostomidis, and Karniadakis 2009, or at the optimal mode interpolation points for the method in Clark, Kutz, and Brunton 2020. This may be adequate for confirming the broad features of a prediction and exploiting the model, but is incapable of exploring oil spill boundaries or unexpected flow shapes and has an intrinsic reliance upon an already accurate trajectory for the decomposition. In-fact, both decomposition and machine learning based approaches are subject to data-fitting, as the time-constraint limits the generation of sufficiently broad data-sets for decomposition or machine learning training. That said, the recent machine-learning based work in Wang, Li, and Chen 2020 shows promise: Although the method in Wang, Li, and Chen 2020 again operates in a reduced order sub-space subject to data-fitting, there is a measure of mode error present.

The complexity (non-linear, stochastic, Lagrangian particles) of the underlying model and sensor placement problem makes solution through global optimisation infeasible within the time-constraint, at least for the full model description with a branch and bound style solver. The research field of optimising with linear, non-linear and stochastic systems, with constraints, is vast and beyond the scope of this review. The reader is directed to Shahriari et al. 2016

and Ehsan and Yang 2019 for an overview. A recent approach has applied the Efficient-Global-Optimisation method to sensor placement to maximise the trace of a vibration modal correlation matrix, with Kriging (or Gaussian-Process-Regression) amongst other approaches used to form a model of the cost function evaluation and thereby simplifying the problem (Morlier et al. 2018). By applying a simpler surrogate model, an update rule on sensor positions that maximises the expected improvement in the surrogate model can be utilised, eventually converging to the optimal sensor locations for the true system. Examples of other methods to produce a tractable sensor placement problem include convex relaxation with local optimisation (Akbarzadeh et al. 2014) and gradient descent style approaches (Joshi and Boyd 2009; Funke, Farrell, and Piggott 2014). These are, in-essence, heuristic and greedy methods.

Returning to the balance of confirmation and exploration measurements. This is analogous to simultaneous estimation and control of the states of a system; mobile sensor-actuators must balance their time between optimal sensing and actuation locations. For brevity, these mobile sensor-actuators will be referred to as sensors here. This problem is addressed in Zammit Mangion, Anderson, and Kadirkamanathan 2011, where an A-optimal (trace of the spatially weighted estimated error covariance) cost function is extended by the squared error between system states and reference states and extended again by a sensor effort term. Such an approach is elegant and intuitive. Mobile sensors therefore display simultaneous exploration and field control behaviour as they minimise this all-in-one cost function, step-by-step to a time-horizon. Results demonstrate intelligent sensor behaviour. A Galerkin decomposition spatial discretisation is used to facilitate efficient evaluation of the cost-function and estimation using a Kalman filter, and the later solving of the cost-function minimisation. Here, the family of spectral PDE solution methods may be difficult to utilise due to the non-linearity, complex bathymetry and system structure of a combined ocean and oil model, combined with the inherent lack of knowledge around a maritime incident. Although the smoothness, regularity and structure of the ocean flow is likely well defined and suitable for a spectral or at least an unstructured mesh solution method, the same can not be said for the distribution of uncertainty. In Zammit Mangion, Anderson, and Kadirkamanathan 2011 the cost function minimisation is recognised as online stochastic optimal control and to make the problem tractable for real-time implementation, a one-step ahead control horizon is utilised though requiring a two-step predic-

tion: One step for sensor movement, another step for sensor affect. Furthermore, to avoid local minima, a grid-based initialisation procedure is employed and sub-regioning acts as velocity constraints on the sensor. Here, it is likely that cost-function evaluation will be prohibitively computationally expensive for a grid-based initialisation and another heuristic may be needed.

2.3.2 Oil monitoring literature

The remainder of this section will discuss the strategic and tactical guidance of sensors, beginning with path planning approaches to target following and moving on to contaminant specific methods in recent literature. Classical path planning encompasses moving to a fixed target location and searching for and following a moving target. Moving to a fixed target bears resemblance to the travelling salesman problem, an NP-hard computationally intractable problem; for which multiple solutions have been developed (Eaton, Chong, and Maciejewski 2016). Broad approaches include:

- A traditional A-star or Dijkstra approach (Dijkstra 1959) has been applied to UAV path finding and planning, due to its easily implementable structure and capability to be adapted to any cell-based cost function (Meng, Gao, and Wang 2009) (Bertuccelli et al. 2009).

- Tabu or Taboo search algorithms (Glover 1990) have also been applied to UAV guidance (Wang et al. 2015), though offer poor performance in comparison to newer techniques and have no handling of constraints or complex objectives.

- Voronoi Diagrams and Discrete Particle Swarm Optimisation (DPSO) (Tong et al. 2012). This work builds upon particle swarm optimization approaches to non-linear function optimization. Particle swarm optimization (PSO) has been utilized in cooperative search UAS path planning (Peng et al. 2009) or path planning for multiple agents (Wang, Li, and Guo 2010) and can include collision-free guarantees (Alejo et al. 2015).

- Genetic algorithms (GA) have been adapted to path planning for both single and multiple UAV and compared to PSO methods, proving to be an improvement (Roberge, Tarbouchi, and Labonte 2013). Further research attempts optimal path generation (Sonmez, Kocyigit, and Kugu 2015) and cooperative planning around tasks using GA's (Geng

et al. 2013).

- Simulated annealing search, using metallurgic processes to solve for global minima. Adapted to UAS path planning (Turker, Sahingoz, and Yilmaz 2015) and compared to a variety of approaches, simulated annealing gave the best performance, but at the highest computation cost. Though its computation gradient was the shallowest, rising the least with increasing problem complexity.

- Discretised control with ant colony optimisation (ACO), employing digital pheromones and adapted to UAS (Duan et al. 2009). ACO and GA, when combined, give improved performance (Shang et al. 2015).

- Receding horizon control (RHC) also known as model predictive control (Mattingley, Wang, and Boyd 2011). Forms a part of many path planning algorithm types, offering control of complex objective, non-linear, constrained, multi-input and multi-output systems. RHC produces future predictions and an input trajectory to follow, at smaller computational cost than global planning method algorithms. RHC has been employed as a decentralized guidance algorithm (Kuwata and How 2007), with safety guarantees (Schouwenaars and Feron 2004) and combined with more abstract algorithms for performance improvements (Xiao et al. 2012).

Algorithms have also been developed specifically for target search and target following with UAVs, employing a large variety of methods. These include underlying employment of, a partially observable Markov decision process (Ragi and Chong 2013), genetic algorithms (Sonmez, Kocyigit, and Kugu 2015), triggered sensor networks (Krishnamoorthy et al. 2012) (Krishnamoorthy, Casbeer, and Pachter 2015) with observation vehicles. Probability density functions with negotiation task assignment framework, for multiple UAS tasking (Moon et al. 2015), demonstrated search capability for multiple UAS's and multiple targets, with minimal overlapping. Receding horizon control with a virtual force modification (Xiao et al. 2012), capable of cooperative search but proved inefficient on irregular search areas. Another method is modified diffusion with a receding horizon and potential method algorithm for coordinated search (Sun and Liu 2009). There are further probability approaches (Bertuccelli and How 2005), pheromone-based algorithms (Qu, Zhang, and Zhang 2015), Gaussian process priors for global

maxima locating (Zhang and Pei 2015) and multiple agent fed probability maps using Bayes' rule (Hu et al. 2013).

Prior path planning research provides multiple satisfactory approaches to guidance generation when searching for simple targets and following them once found. However, some lack consideration of the target's dynamics and rely upon an initial guess at the target's location, offering little in the context of maritime contaminants where the information available at the start (an observed slick of oil) has no guarantee of containing the source location or providing a trail to the source. The algorithms are not focused on continually developing maritime situations, where the target number of oil slicks is not known, targets must be followed but not at the expense of area mapping for other targets and there is both time-dependent and location-dependent importance for each target.

In the oil industry, for monitoring oil spills there exists a variety of models (Spaulding 2017) used to predict the spill trajectory, and further variation in definitions of spill uncertainty (Goncalves et al. 2016), and decision support systems to evaluate potential responses (Nelson and Grubesic 2019). However, existing models and decision systems often confine their flight patterns to standard profiles, focus solely upon spill sensing and do not support real-time data assimilation of measurements to correct utilised large-scale fluid model data. For an example of strategic guidance, as in decision support systems, see Ye et al. 2019, where a simulation-based multi-sensor particle swarm optimization approach assigns resources to monitor and clean-up an oil spill. Oil spills were assumed static and, as the beginnings of a common theme in the literature, there is no feedback loop present between the sensors or clean-up skimmers, and the model predicting spill location and clean-up time. Hence it is unclear what their approach would do if, upon arrival at a supposed location, no oil spill is present. If it assumes there is no spill and declares the mission complete, this is an incorrect response. Instead, there needs to be an appropriate sensing strategy to correct the original spill trajectory prediction.

A consistent thread through all of the oil monitoring control literature is the limiting of sensor consideration to determining an accurate now-cast, or an accurate source-location, when monitoring an oil spill. With very few exceptions, which are discussed later in this section, there is no consideration of utilising measurements to improve the accuracy of the environment model. Given the variety of information required in spill responses and litigation, there should

be a focus upon increasing estimation accuracy of the spill trajectory across the whole spatio-temporal domain. In support of this, the monitoring framework must be capable of establishing an accurate flow model for the region, with the directing of suitably equipped sensor platforms to measure fluid velocities at crucial locations. The placement of sensors to determine fluid flow and the principal dynamic structures within a flow field is a whole research field in itself, beyond the scope of this Chapter. The reader is directed to Yildirim, Chryssostomidis, and Karniadakis 2009, Annoni et al. 2018, Clark, Kutz, and Brunton 2020 and Jayaraman and Al Mamun 2020, which contain modern decomposition based approaches, similar to one evaluated and ultimately discarded and replaced in this work, further detailed in Chapters 5 and 6.

The most closely related work in the sensor control field, that focus upon tracking oil spills or drifting objects, utilise an underlying model to generate oil or object predictions over time. However these models are extremely simplistic, using uniform flow in small areas, with constant diffusion coefficients and no environmental constraints. Furthermore, the models themselves are underutilised; for example, merely being used to generate a sinusoidal flight path around the supposed oil spill edge (Zhang and Pei 2015). There has been application of model-based optimisation to oil spill clean-up trajectories (Kakalis and Ventikos 2008; Grubesic, Wei, and Nelson 2017), and several multi-agent sensor approaches have followed bio-mimetic approaches in swarm behaviour (Banerjee, Ghosh, and Das 2018; Bruemmer et al. 2002) to track oil spills and further work uses cost-function minimisation to plan samples (Yan et al. 2018). Yan et al. 2018 and Lian et al. 2018 minimise the mean-squared error between the estimates of a Gaussian-Process-Regression model and sensor measurements, map a spill, and explore a domain through a three-stage algorithm. Closely related work in the atmospheric field plans a sensor path that maximises a utility function based on the estimated information gain of each measurement in a Bayesian framework, seeking to determine the source parameters (Hutchinson, Liu, and Chen 2019). As noted in Hutchinson, Liu, and Chen 2019, the presence of turbulent flow, contaminant irregularities and sparse sensing limits the effectiveness of gradient-based approaches for oil spill mapping. In prior methods, though some do utilise a model of contaminant spread, their models lack the oil dynamics, cyclical tidal flow, or wind, wave and water coupling necessary to capture the key dynamics in oil spreading. For example, Hutchinson, Liu, and Chen 2019 nicely incorporates the effect of underlying flow on their tracer when sensor path planning,

through generation of an expected observation using their model, but assumes a constant mean flow velocity. In this work, it is acknowledged that a significant source of uncertainty is the underlying time-varying velocity fields and the definition of uncertainty is structured to incorporate this. Hence, improvement in estimation of the environment is considered when planning sensor paths to minimise uncertainty.

Use of information theoretic measures of uncertainty minimisation is present in uncertain environment path planning literature (Feder, Leonard, and Smith 1999; Candido and Hutchinson 2011), and some cognitive search strategies (Vergassola, Villermaux, and Shraiman 2007; Hutchinson, Liu, and Chen 2019) are formulated as an optimisation using a partially observable Markov decision process (Chong, Kreucher, and Hero 2007), though these focus on source term estimation. For example, Vergassola, Villermaux, and Shraiman 2007 minimises the expected entropy of the posterior probability map of source location.

2.3.3 Summary and further considerations for sensor guidance

To effectively summarise the review of control literature for oil spill monitoring, particular attention is given to four papers: These are Zammit Mangion, Anderson, and Kadirkamanathan 2011, Hutchinson, Liu, and Chen 2019, Yan et al. 2019 and Pashna et al. 2020. The strengths of Hutchinson, Liu, and Chen 2019 and Yan et al. 2019 are their stochastic underpinnings and optimisation: Through the use of a Bayesian framework and Gaussian-Process-Regression, information theoretic measures form a cost function to be optimised by sensor guidance. In Hutchinson, Liu, and Chen 2019, this is maximising the utility of the expected measurements in the next update cycle. In Yan et al. 2019, this is minimising the variance within the Gaussian-Process-Regression model, and variance of the observation prediction value of GPR regression model. The major features that stand-out above other work is their consideration of enhancing the accuracy of their underlying model and forming of an all-in-one optimisation problem that does not require multiple control approaches. However, both Hutchinson, Liu, and Chen 2019 and Yan et al. 2019 employ a simplistic model. While Hutchinson, Liu, and Chen 2019 states it is possible to substitute any model into the Bayesian framework (this includes Gaussian-Process-Regression, Kalman Filters/Smoothers etc), a stochastic 2.5D hydrodynamic and oil particle model with 10^5 variables would considerably complicate the propagation of variable

distributions and uncertainty between time-steps. Hence some abstraction of the model will be required. In Pashna et al. 2020, an oil spill monitoring framework is suggested that makes use of a more complex model, described by a stochastic PDE with weathering included. However, this models 2D surface dynamics only, without wave processes, entrainment or buoyancy. Furthermore, though a hybrid Fuzzy and Artificial Potential Field controller demonstrates an approximately 85% precision in tracking an oil spill, once again there is no consideration of enhancing the accuracy of their underlying environment models. Finally, the cost function extension and solution method in Zammit Mangion, Anderson, and Kadirkamanathan 2011 demonstrates an approach capable of multi-objective sensor control; to both measure and remove uncertainty at peak locations, but also explore the domain to improve the model.
Note that the above identified papers are both closely related and very recent. The work here-in is similar in some aspects: The underlying model contains a stochastic PDE, but in 3D/2.5D for oil and environment flow respectively and makes use of a linear wave model. This is a significant step towards a more accurate and complex model to utilise in a control algorithm. A constrained model-based optimisation problem is formulated and solved, that contains information theoretic measures including variance, probability and entropy. There is also emphasis on measurements that enhance the accuracy of extrapolation of an oil spill forward and backwards in time.

2.4 Data assimilation

Sensor data, assumed to be point measurements of states, must be used to estimate the states of the entire environment flow fields, wind and ocean, as well as updating the oil particles. Due to the complexity and the high numbers of states in a combined ocean and oil model, a full-state estimator is infeasible. Commonly used methods in the hydrodynamic field include the SEEK filter, see Rozier et al. 2007, Ensemble Kalman-Filters/Smoothers(EnKF/EnKS) extended in Raanes, Bocquet, and Carrassi 2019 and 4-Dimensional Variational Assimilation (4D-VAR) (Amezcua, Goodliff, and Van Leeuwen 2017). Note there are also hybrid methods (Cessna, Member, and Bewley 2010). The SEEK filter is also known as a reduced-order Kalman filter and under certain conditions, and relaxation of the Gaussian distribution of states/observations assumption, is akin to the reduced order information filter (Chin, Haza, and

Mariano 2002). SEEK is concisely described as an Extended Kalman Filter (EKF), where the error covariance matrices are truncated to a reduced low-order space and propagated through time in their reduced form. Ensemble Kalman methods utilise a collection of simulations forming an ensemble of propagated perturbed state trajectories to implicitly approximate the propagation through time of an error covariance matrix. 4D-VAR is a space/time variational minimisation of state error, similar to Moving Horizon Estimation in the controls community (Michalska and Mayne 1995), that also requires estimation of covariance matrices but also tangent linear approximations of the system to the first order (Jacobian matrix) and potentially second order (Hessian matrix) and the adjoint of the system dynamics. 4D-VAR without modification, operates directly on the high-order model. Substitution of reduced order models and approximations into 4D-VAR, or their approximation through an ensemble (discussed in Amezcua, Goodliff, and Van Leeuwen 2017), alongside Hessian-free (Daescu D. N., Navon 2006) solver methods, has attempted to reduce the computational complexity of 4D-VAR. Methods of reducing the order of the system description and size of the associated covariance matrices vary, with Proper-Orthogonal-Decomposition (Daescu D. N., Navon 2006), Dynamic-Mode-Decomposition (Iungo, Abkar, and Port 2015) and Empirical-Orthogonal-Functions (Rozier et al. 2007) in use amongst others.

Parallel model runs or downloading multiple external data sets is not necessarily feasible in remote environments, where only a singular operational model may be available, if at all. Furthermore, even if multiple operational models are available, unless there is a high fidelity small scale model also available, the data-sets may be large-scale and very similar. Hence a formed ensemble may not include worthwhile perturbations that capture the variance induced by small eddy currents or shipping traffic. Therefore, while ensemble based methods of estimating trajectories and covariance matrices may still be a useful, it is a trade-off against the extra cost of data-access and computation resources.

Needed for full-order and accurate variational assimilation, tangent linear descriptions of a stochastic Lagrangian oil spill are impractical due to random variables and ever changing state numbers, especially when coupled with an Eulerian fluid model. Although tangent-linear approximations and reduced order implementations of variational assimilation could be viable, the advantage that variational assimilation has model-described knowledge of system behaviour

may be lost as the solver must operate in the limited subspace of the reduced-order description.

2.5 Summary

Condensing the literature reviews above, there is potential for application of mobile sensors upon UAVs to monitoring contaminants. The models currently used to predict spill trajectories are varied and either independent from a hydrodynamic model and fast, or coupled to a hydrodynamic model and slow. There is therefore an incentive to develop a highly streamlined combined ocean and oil model, that trades absolute accuracy for a fast computation speed for a control feedback loop, with the ability to assimilate both oil and fluid measurements. The developed model should be integral to the chosen guidance algorithm and thus improve upon the models currently utilised in the control field. The model-based guidance algorithm should consider model improvement, environment and oil spill estimation, analysis and prediction, in an all-in-one solution. Finally, a data-assimilation method that is ensemble-free and tangent-linear model-free is required, due to the computation requirements and system description respectively. Together, the combined ocean and oil model, model-based sensor guidance and data assimilation methods form a model-based adaptive monitoring strategy.

Chapter 3

The Adaptive Monitoring Framework

This chapter provides a brief overview of the proposed adaptive monitoring framework, with the following chapters providing detailed explanation of each of the key stages.

The adaptive monitoring framework is shown in Figure 3.1. At the heart of the framework is the fluid and oil model, the Sheffield Combined Environment Model (SCEM) (Hodgson, Esnaola, and Jones 2019). At each time step, SCEM solves the wind velocity, ocean velocity and oil particle velocity at each grid point within a discretised spatial domain, and also evaluates a probabilistic function of oil particle presence. These are combined into a state vector and solved forwards in time, over a specified time horizon, to produce a state trajectory. This constitutes the Stage 1 of the framework. The computed state trajectory is subsequently used separately to solve an optimisation problem (Stage 2) and identify a reduced order model of the plant (Stage 3). The solution of the optimisation problem returns the optimal sensing locations for minimising a function of uncertainty in oil spill particle location. This is solved to a time horizon, thereby providing each mobile sensor with a path trajectory. The sensors navigate to the first location along this path and take a measurement of the ocean surface velocity, wind velocity and oil thickness, at that point in space. The measurements are utilised with the identified reduced order model, in an estimation stage (Stage 4). The estimated states of the reduced order model are mapped back to the physical states and used to re-initialise SCEM at the next time step. This sequence is recurrent in a receding horizon fashion.

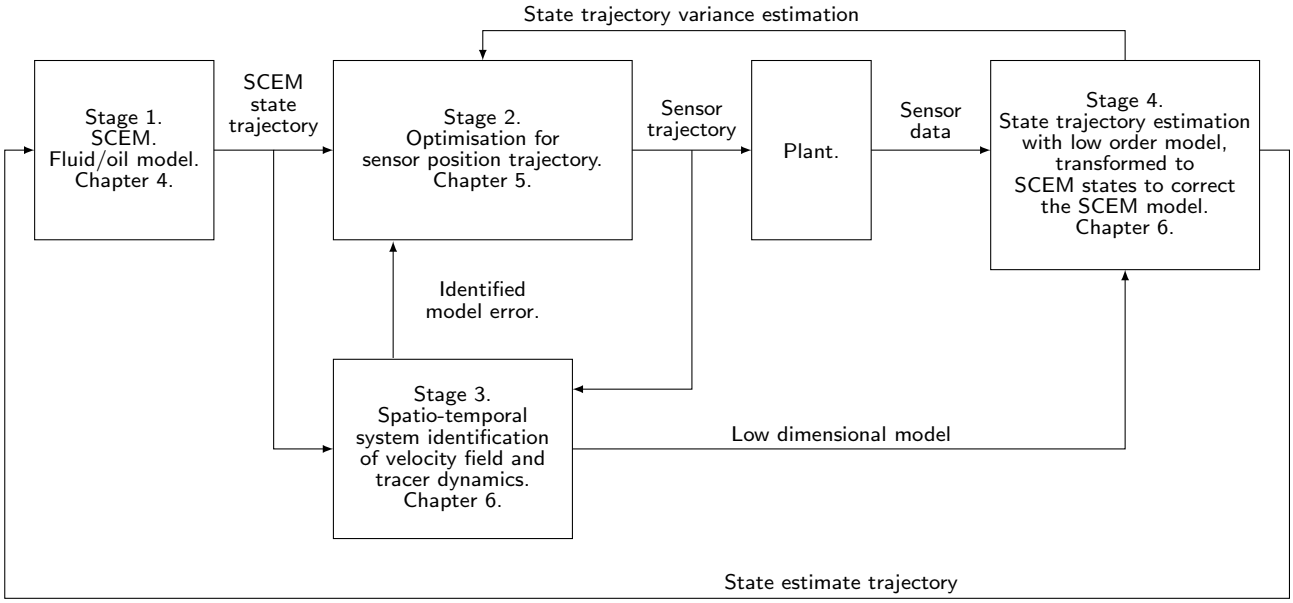


Figure 3.1: A block diagram of the framework for adaptive monitoring using SCEM, demonstrating the system feedback.

3.1 Overview of methods

This section provides an overview of the methods utilised in each stage of the framework.

Stage 1 (Chapter 4) has a 2D Navier-Stokes solver, to resolve low-fidelity ocean and wind flow around the local geography. The solver is similar to the work in Stam 2001, using finite differences to approximate spatial derivatives, with forward Euler explicit time-stepping of advection and backwards Euler implicit time-stepping of diffusion, with a varying time-step. Empirical approximations extrapolate the ocean surface and wind flow velocities to an ocean sub-surface velocity profile, including wind-shear flow, tidal flow and Ekman currents, and a linear wave model uses wind and ocean flow to produce an ocean wave spectrum. The oil model component of Stage 1 utilises a standard Langrangian particle approach, where oil particles are moved around by the ocean and wind dynamics, with minor corrections and modifications from recent literature. Chapter 4 also includes a Fokker-Planck type equation (Risken 1996), where instead of a PDE description of a particle ensemble's position probability density function, a pair of PDEs describe the evolution of a particle ensemble's position variance and produces a scalar measure of uncertainty. The scalar measure being the position uncertainty (as an area) of particles at a spatio-temporal location. These uncertainty PDEs use a further Stam-style discrete solver, in a state-space formulation for the application of state space optimisation solvers in the next stage and chapter.

Stage 2 (Chapter 5) finds a sequence of sensor measurement positions that best inform the model of oil particles and fluid flow, up to a future time. To do so, an uncertainty minimisation optimisation is formulated, with the sensor position trajectory to a receding horizon being the optimisation variable. The uncertainty in this optimisation is governed by the the uncertainty PDEs described in Chapter 4. This is a locally-convex, non-linear optimisation problem with time-varying constraints on sensor position. The optimisation is solved for an optimal sensor position trajectory using a gradient descent method. The gradient of the optimisation cost function, with respect to the sensor positions, is determined through application of the discrete Adjoint method. The Adjoint method uses the dual form of the constrained optimisation problem to form a backwards recursion algorithm, that introduces and solves for a new set of Adjoint variables backwards in time. These Adjoint variables are then used to determine the cost function gradient.

Stage 3 (Chapter 6, Section 6.1) uses the state trajectory (data set of state snapshots over time) of Stage 1 to form a reduced order model. While the non-linear model in Stage 1 utilises a large number of states (several thousand) to describe fluid flow, the linear reduced order model might have just a few, where each reduced order state describes a large flow structure. For example, one state might describe the amplitude of the main flow in a river channel (See Figure 3.2), and another the amplitude of the main eddy swirl at a river mouth. The reduced order model adjusts the amplitude of these flow structures with time to capture the dynamics of the flow. Hence the reduced order model describes the the large state trajectory using a few mode shapes, the initial condition of the mode amplitudes, and changing of the mode amplitudes with time. The method of model reduction extended in this thesis is Dynamic Mode Decomposition (DMD), a technique to extract spatio-temporally coherent structures, and their dynamics, directly from the high dimensional data of the SCEM state trajectory. DMD forms a set of modes, each with a fixed oscillation frequency and decay/growth rate. Each mode has its oscillatory temporal behaviour contained in a mode dynamics matrix, analogous to a discrete time linear-time-invariant state-space dynamics matrix. See Figure 3.3 for an example of the mode dynamics matrix eigenvalues. The DMD modes are not orthogonal in space (like Principal Component Analysis), but instead groups spatially distributed states into modes according to their sinusoidal behaviour in time. This is particularly useful when performing dimensional

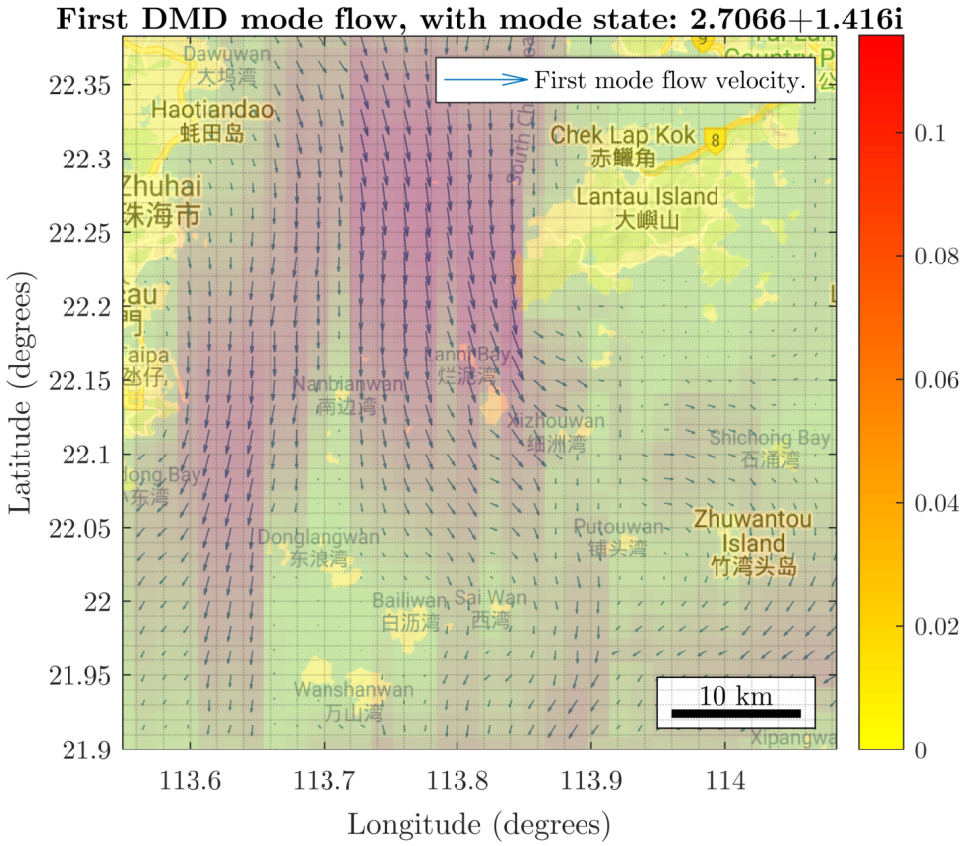


Figure 3.2: The mapping of the first DMD mode to tidal flow around Hong Kong, with a mode state value of $2.7066 + 1.416i$. As the first mode, this is the mode that captures the most energetic structures in the flow and hence describes the Zhujiang river.

reduction on an oscillatory data-set, such as those involving tides, as a measurement in one spatial location can be applied across the domain to all locations that are strongly coupled in time through estimation of the underlying mode amplitudes.

Stage 4 (Chapter 6, Section 6.1.1) uses measurements of flow velocity to estimate the mode amplitudes of the reduced order model and hence the flow of the entire domain. The mode estimates can be stepped backwards or forwards in time, using the linear reduced order model dynamics, to produce an analysis or predicted flow trajectory that is resolved to a higher fidelity in a rerun of the high order model SCEM, where SCEM is driven by the reduced order model trajectory and velocity measurements. In addition to estimating the flow velocity using sensor data, after several measurements are gathered, the estimates of earlier flow velocities are adjusted (or smoothed) for better accuracy using the later sensor data. The selected method of reduced order model state estimation is the Rauch–Tung–Striebel (RTS) smoother, a two-pass algorithm for fixed interval smoothing. The forward pass is a time-varying Kalman Filter (KF): Despite the reduced order model being linear-time-invariant the reduced order model is

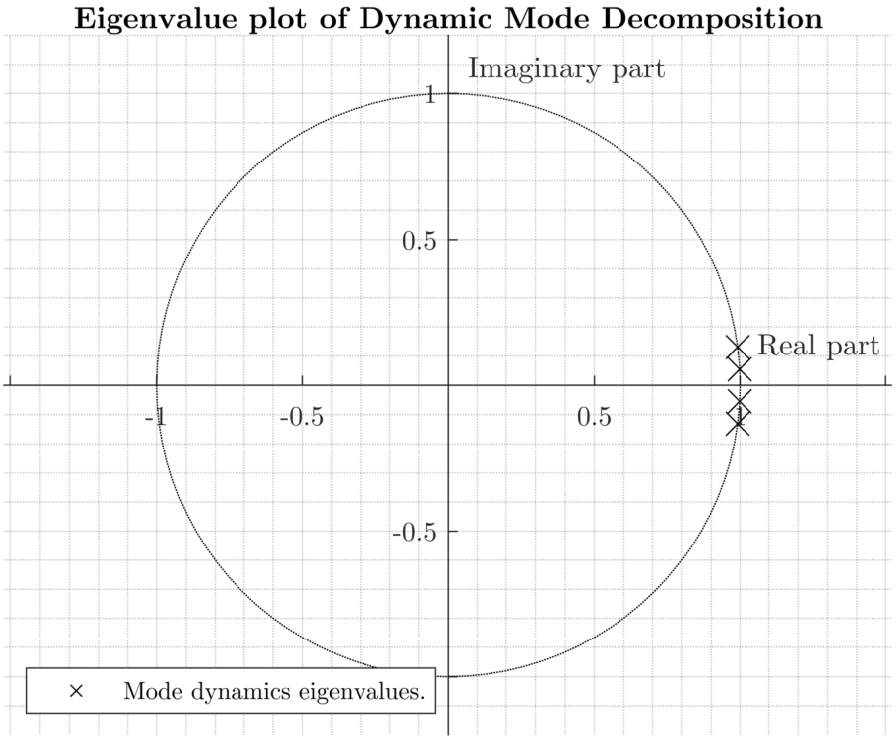


Figure 3.3: The reduced order model eigenvalues for a four mode reduced order system. Note that all eigenvalues are Lyapunov stable but un-damped, being exactly upon the unit circle, and hence oscillate with the same amplitude indefinitely. These eigenvalues have oscillatory periods of approximately 12 (larger imaginary component) and 24 hours (smaller imaginary component). This is expected for an asymmetric tide cycle, where the sinusoidal profiles comprise of one standard 12 hour period oscillator to describe the tides, and a longer 24 hour period oscillator that alters the amplitude of the first and second tides when the oscillators are combined.

redetermined regularly, so a non-linear Kalman filter is required to handle the changing system. A Kalman filter is an efficient optimal estimator, utilising a series of noisy sensor measurements, a model of the estimated process and a recursive methodology to produce state estimates more accurate than estimates based on a single measurement alone. The Kalman filter also estimates a measure of the state estimate uncertainty. The backward pass is a further recursive algorithm that steps from the latest estimate to the earliest estimate, utilising the forward pass Kalman filter variables and measurements to smooth the state estimates, also adjusting the smoothed state estimate uncertainty.

3.2 Monitoring framework iteration

The monitoring framework is an iterative process, with an initial forward run being used for analysis, providing conditions for another forward run and further analysis and so on, towards a receding horizon. During a forward run the reduced order, DMD model, is re-calculated whenever new measurement information is available. Each time there is new sensor data, the reduced order model is again determined from the moving time-window of the state trajectory and has amplitudes estimated by the RTS smoother. This smoothed trajectory is an analysis trajectory. The RTS smoother uses prior flow estimates and external data as measurements, and sensor data if sensors are active. By stepping the Kalman filter component of the RTS smoother forward, ahead of the current time, the reduced order DMD model can be used to predict future fluid flow and can assimilate external data predictions if desired. The prediction of later (e.g an hour ahead) fluid flow, oil movement and uncertainty uses SCEM, but forced by the reduced order model fluid prediction and Kalman filter. Hence sensor path planning is informed by prior sensor measurements. Once an analysis point has been reached, an analysis trajectory is calculated, see Section 6.1.1. This analysis trajectory is then used to force another forwards run of SCEM. A framework time-line for the simulations of Chapter 7, Section 7.2 and Section 7.3 is provided in Figure 3.4.

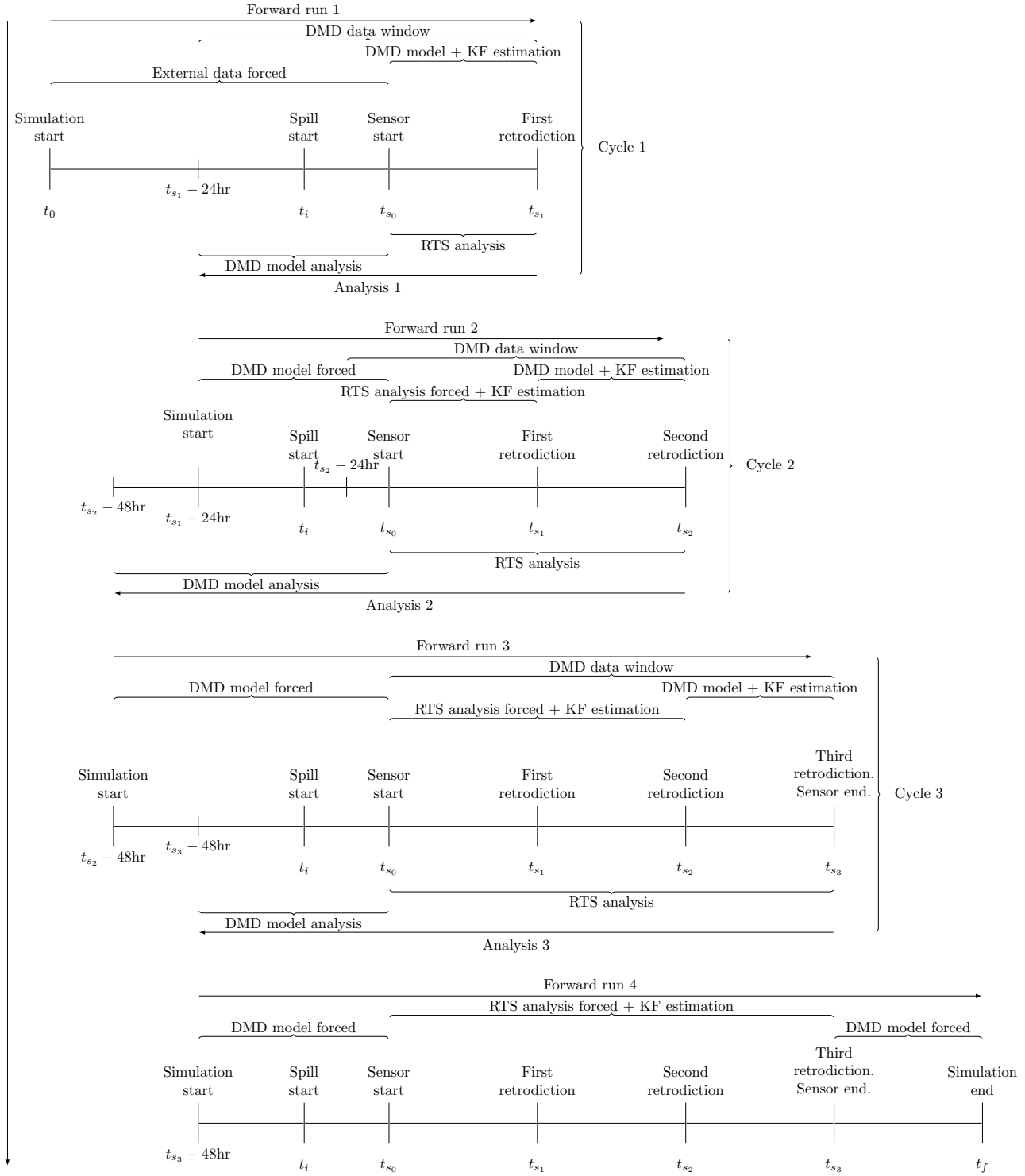


Figure 3.4: A time-line displaying the iterative refinement of the model estimates, through alternating forward and analysis passes. Forward runs of SCEM are initially forced by external data and then through DMD and Kalman filter estimation. After analysis, which can include source location estimation, forward runs are forced through the DMD model and RTS analysis trajectory, with Kalman filter estimation if sensors are active. For Section 7.2, the analysis times t_{s_1}, t_{s_2} and t_{s_3} are after 12, 20 and 24 hours of sensing respectively.

Chapter 4

Modelling (Stage 1)

This chapter first presents the Sheffield Combined Environment Model (SCEM), for the purpose of providing online control guidance to assets with minimal supporting data. The SCEM model is described, giving equations and algorithm in both flow chart (Figure 4.1) and pseudocode (Algorithm 1). The model is then demonstrated to accurately predict a real-world oil spill in the Bay of Biscay 2019 and give similar results to an industry standard oil model GNOME when given the same input data for a spill near Hong Kong 2019. Next, a suitable description of oil spill uncertainty is developed from the equations used to model oil processes in SCEM.

To reiterate an earlier statement, the Sheffield Combined Environment Model does not present a revolutionary step in the fields of hydrodynamic simulation or oil modelling, but instead provides a fast, combined model for guidance of mobile sensors. This model is however, a step forward for the model-based control field. Some components of the model are described only in high-level detail, as referenced work offers a wealth of information. Any cases where there is an unconventional description of a process is related to work in future sections. For example, describing the turbulent diffusion process as a random walk, rather than incorporating it into a complete description of particle movement as a stochastic PDE is to maintain clear separation of the stochastic models for each process, for later analysis.

It is important to note that this model utilises external data provided by large-scale, complex simulations of boundary layer wind and ocean flow as boundary data and estimated values if available. The flow described by the external data is resolved around the local geography that may not be present in the larger-scale external models. This is similar in purpose to the CATS model (Galt 1984) employed by NOAA to inform the GNOME model (NOAA 2012). Flow is

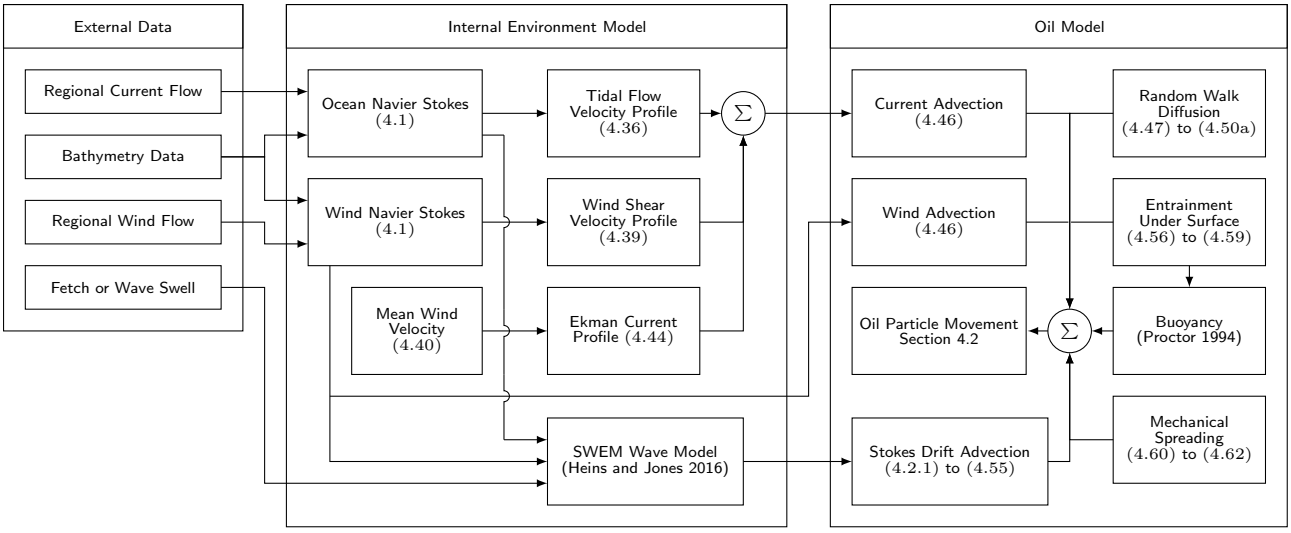


Figure 4.1: A block diagram of the combined fluid and oil model, showing the initialisation with external data and the coupling between wind, current and wave motion in producing a contaminant velocity field.

resolved using the 2D incompressible Navier-Stokes fluid equations, which are reasonable to use at the air-water interface where surface fluid density is approximately constant on a maritime incident scale (e.g 50 km by 50 km).

4.1 Environment modelling and model structure

The environment model contains interconnected sub-systems that describe local ocean currents, local Ekman currents, local wind and local wave conditions. Figure 4.1 shows the main physics sub-components and their interactions. At each time-step the local wind field is calculated first, followed by the local ocean current velocity field, then the depth velocity profiles are calculated and finally the wave model is updated to produce a wave induced velocity. These are used, together with oil-only effects such as turbulent diffusion, mechanical spreading, entrainment and buoyancy, to move oil particles. The complete forward simulation algorithm is described in pseudocode in Algorithm 1.

4.1.1 Domain structure

The spatial domain is denoted by $\Omega \subset \mathbb{R}^3$, and represents a cuboid section of the Earth including land and ocean with a given depth. The upper surface, at the air to water/land interface, of the domain is $\partial\Omega \subset \mathbb{R}^2$. The surface is discretised into a regularly spaced grid of $n_x \in \mathbb{N}$ grid cells

(west to east) and $n_y \in \mathbb{N}$ cells (south to north), with spacings $\delta x \in \mathbb{R}_+$ and $\delta y \in \mathbb{R}_+$ in the respective directions. A grid cell at indexed position (x_i, y_j) covers the Cartesian coordinate positions: $(x_i \pm \frac{\delta x}{2}, y_j \pm \frac{\delta y}{2}) \subset \partial\Omega$, where index $i \in [\![1, n_x]\!]$ is the west to east horizontal grid index, where $j \in [\![1, n_y]\!]$ is the south to north grid index. Continuous time $t \in \mathbb{R}_+$ has a corresponding discrete time $t_k \in \mathbb{R}_+^{n_t}$ with a time-step index $k \in [\![0, n_t - 1]\!]$ for $n_t \in \mathbb{N}$ varying time-steps of $\delta t : \mathbb{R}_+ \to \in \mathbb{R}$. A particular time-step size at discrete index k is $\delta t_k : \mathbb{N} \to \mathbb{R}$. The trajectory of t_k, formed by column stacking (or concatenating) t_k for all k, is $\mathscr{T} \in \mathbb{R}^{n_t}$. The initial time is $t_0 \in \mathbb{R}_+$ and $t_f \in \mathbb{R}_+$ the final time.

Subsurface water is discretised with a two stage fine and coarse mesh, such that for each grid cell there exists a set of depths $\mathbf{z}(x_i, y_i)$, defined by

$$\mathbf{z}(x_i, y_i) = \{0, \delta z_1, 2\delta z_1, ..., N_{\text{crit}}\delta z_1, z_{\text{crit}}, z_{\text{crit}} + \delta z_2, z_{\text{crit}} + 2\delta z_2, ..., N_{\bar{z}_{ij}}\delta z_2 \bar{z}_{ij}\}.$$

Depth spacings $\delta z_1 \in \mathbb{R}_+$ and $\delta z_2 \in \mathbb{R}_+$ are the finer and coarser vertical grid spacing respectively, $N_{\text{crit}} \in \mathbb{N}$ is the number of fine mesh grid cells. The switch depth from fine to coarse mesh, $z_{\text{crit}} \in \mathbb{R}_+$, is determined by the maximum depth of oil particle insertion into the water column (explained in Section 4.2.2), or specified by the user. By utilising a two stage depth grid, finer detail can be maintained near the surface where the majority of contaminant mechanics take place. A 3D grid cell is specified by the indexes (x_i, y_j, z_w), where $w \in \mathbb{N}$ is the surface to sea floor grid index.

Note that a regular grid was selected in place of unstructured or spectral alternatives: In the absence of prior information, or in the presence of often poor available information about the maritime incident, there is no information available on the smoothness of the spatial gradients of the fluid flow, or oil spill uncertainty, across the domain. Hence a regular grid of appropriate size, balancing computation time with fidelity requirements, ensures a solution throughout the region. Grid size in SCEM is determined by the sensor swathe width and mobility constraints, data assimilation time spacing and computation time needed.

Grid spacing

An example of grid spacing determination is now given, suitable for the simulations of Chapter 7. Consider a UAV capable of flying at 60 mph, equipped with a 1920 x 1080 pixel optical

sensor with a 30° field-of-view. These are approximate figures for an inexpensive UAV with a telephoto-lens equipped High-Definition video camera. The UAV is required to supply thickness and volume information, and hence requires an oil sensing resolution of 0.5 m for a small spill, (see Table 2.2). To achieve a spatial sensor resolution of 0.5 m, the UAV would fly at an altitude of 2000 m, with a 1000 m sensor swathe. Given the 1000 m sensor swathe, a grid cell size of 1000 m by 1000 m is appropriate: Assuming an optical flow tracking capability, the mean of the sensor flow observations can be taken for the grid cell flow, while Langrangian oil particles can be added, removed, or modified as appropriate within the observed cell. The World Meteorological Organisation require a temporal resolution of ≤ 1 hour for oil monitoring (Golding et al. 2001), but due to the complex domain and environment (busy shipping lanes, strong asymmetrical tides) a 15 to 30 minute temporal resolution seems prudent in Chapter 7. At a 1000 m by 1000 m grid cell size, with a 15 to 30 minute time-step, the framework is capable of operating sufficiently quickly for real-time UAV control and data assimilation.
Note that the reduced order DMD model is far lower fidelity and filters out high frequency components of flow. One reduced order state might describe the flow of an entire river channel for example. However, the low-order projection is resolved to a higher accuracy by the 2D fluid model in SCEM, and sensor measurements inject additional fidelity in important regions.

Model states

The combined ocean and oil model described here is a component of the larger adaptive monitoring strategy. As such, it contains many internal states that are not needed in the full system. In SCEM, each grid-cell is defined by its geo-spatial coordinates and contains the following states:

- Environmental information (temperature, water density etc).
- Wave spectra.
- Current time wind velocity.
- Previous 12-hour mean wind velocity.
- Tidal flow velocity profile.

- Wind induced surface shear flow velocity profile.
- Ekman current velocity profile.
- Stokes drift velocity profile.
- Probability of oil presence.

The states are formally defined in the following subsections. Note that water density is assumed constant through the water column, justified by the focus of this work upon surface oil slicks.

System states

The state vector passed to the monitoring framework is $\vec{x} : \mathbb{R}_+ \to \mathbb{R}^{n_{\text{cell}} n_x n_y}$, containing the cell centred states. The number of states per grid cell is denoted by $n_{\text{cell}} = 7$. The seven states in the state vector are the horizontal components of surface wind, ocean and oil drift velocities and a further state related to the probability of oil presence. The further states in SCEM, describing the wave spectrum, other velocity profiles and oil properties are only used as needed, where oil particles are present in the simulation and are absent from the general system. Define the total number of states as $n_s \in \mathbb{R}$, where $n_s = (2(n_x + 1)n_y) + (2n_x(n_y + 1)) + (n_{\text{cell}} - 4)n_x n_y$. As a staggered grid, horizontal velocities have an extra column of states, while vertical velocities have an extra row of states.

4.1.2 Flow solver

A 2D Navier-Stokes solver has been implemented to determine local flow velocities for both wind and water, using assumed, measured or external model-provided boundary data. The general form of the 2-dimensional incompressible Navier-Stokes equations are:

$$\frac{\delta \vec{U}}{\delta t} = -(\vec{U} \cdot \nabla)\vec{U} + \nu \nabla^2 \vec{U} - \nabla p + \vec{s}_U, \tag{4.1a}$$

$$\nabla \cdot \vec{U} = 0, \tag{4.1b}$$

where $\vec{U}(x, y, z, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^2$ is the in-plane velocity field such that $U(x, y, z, t) = [u(x, y, z, t), v(x, y, z, t)]^{\mathrm{T}}$, with $u(x, y, z, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}$ and $v(x, y, z, t) : \Omega \times \mathbb{R}_+ \to$ the

in-plane velocity components in the west to east and south to north directions respectively. For notational brevity the space and time dependency of variables is not shown in subsequent equations. In (4.1), $\nu \in \mathbb{R}_+$ is the kinematic viscosity of the fluid, $p(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ is the surface internal pressure field and $\vec{s}_U \coloneqq \vec{s}_U(x, y, t) : d\Omega \times \mathbb{R}_+ \to \mathbb{R}^2$ are external surface forces, if present. Also in (4.1), $\nabla\cdot$ and $\cdot\nabla$ are the divergence and directional derivative operators respectively. For wind flow $\vec{U} \coloneqq \vec{U}_{\mathrm{w}} = [u_{\mathrm{w}}, v_{\mathrm{w}}]^{\mathrm{T}}$, for ocean current flow $\vec{U} \coloneqq \vec{U}_{\mathrm{c}} = [u_{\mathrm{c}}, v_{\mathrm{c}}]^{\mathrm{T}}$ and for Ekman wind $\vec{U} \coloneqq \vec{U}_{\mathrm{E}} = [u_{\mathrm{E}}, v_{\mathrm{E}}]^{\mathrm{T}}$. Flow is determined for ocean surface currents and for wind velocities at 10 m above sea level by solving (4.1) subject to spatio-temporal boundary conditions upon the velocity. These are set from external data, measured data, or by setting $\vec{U}(x, y, z, 0)$ to a best-estimate of mean flow if no data is available. Boundary conditions are described in the next section.

The Navier-Stokes equations are spatially discretised upon a staggered grid, with spatial derivatives approximated by finite differences (F. Harlow and J. Welch 1965). The staggered grid structure is displayed in Figure 4.2.

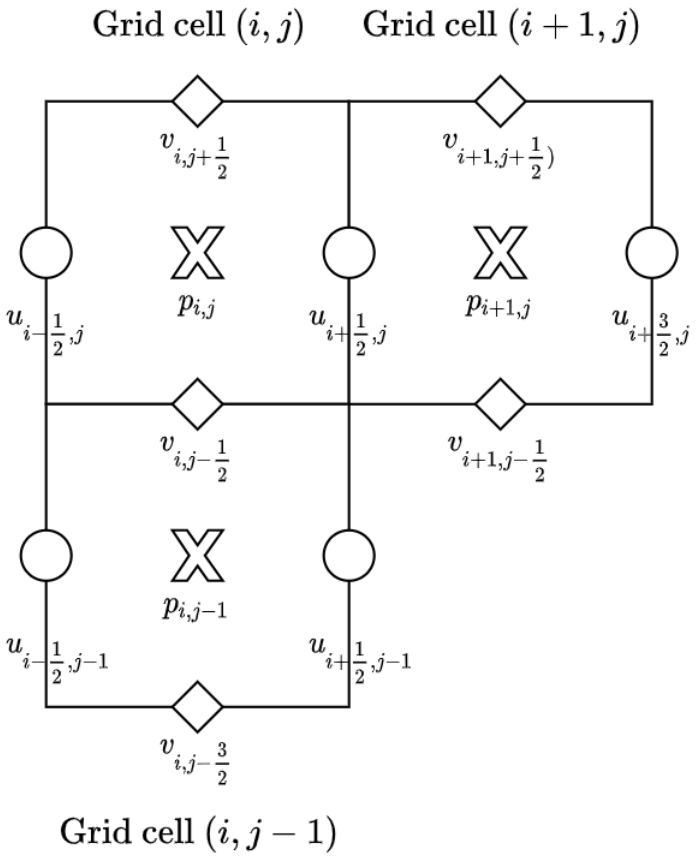


Figure 4.2: This figure displays the grid structure of the fluid solver, note the velocities on cell boundaries, while pressure is defined at the cell centres. This grid structure is shared by the uncertainty PDE discretisation, where the uncertainty state replaces pressure at the grid cell centre. Where velocities or uncertainties are required at spill centres, the mean is taken of surrounding values.

With respect to time-stepping, diffusion terms are solved using a backward Euler method and Gauss Seidel Successive Over Relaxation (Stam 2001), whilst advective terms are solved using 1st, 3rd or 4th order Runge-Kutta methods. Mass conservation is enforced via an iterative pressure projection step, in which the pressure field is found using Gauss-Seidel Successive Over Relaxation (Stam 2001), with subsequent correction of the velocity field. The pressure field is determined, the velocity corrected, then these steps are repeated $2n_\phi$ times, where $n_\phi \in \mathbb{N}$, using successively corrected velocity fields until the flow-field divergence is below a tolerance.

Time stepping

The time-step δt is shared between the hydrodynamic model, the oil model and the later uncertainty model. The time-step is variable with the step size determined by the Courant

number (Courant, Friedrichs, and Lewy 1967), or by the vertical diffusivity coefficient (Visser 1997), or by a user specified value, whichever is smaller. In practice, this would invariably be the Courant number determined step size, but a maximum step-size of 15 to 60 minutes is recommended to improve the temporal resolution of results for analysis. In the simulations of Sections 4.4 and 7.2, a user specified time-step limit of 30 and 15 minutes respectively effectively implemented a regular time-step for each simulation, as the user specified time-step remained smaller than the Courant number and vertical diffusivity determined time-steps.

Boundaries, measurements and obstacles

Obstacles are regions of $\vec{U}_{\mathrm{c}} = 0$ for ocean current flow velocity, or $\vec{U}_{\mathrm{w}}$: $\left\|\vec{U}_{\mathrm{w}}\right\|_2 \leq \kappa^2 \left\|\vec{U}_{\mathrm{w}_{\max}}\right\|_2$ for wind flow where $\kappa \coloneqq \kappa(x, y) : \partial\Omega \to \mathbb{R}_+$ is a wind resistance coefficient based on the environment and $\vec{U}_{\mathrm{w}_{\max}}$ the maximum wind velocity. The presence of obstacles, such as coastline geography, is accounted for by the use of Dirichlet boundary conditions on the velocity field in relevant grid cells. Due to the staggered grid implementation, this is a form of semi-slip boundary (F. Harlow and J. Welch 1965). This is not unprecedented in ocean models, a user selected value for slip is found in the NEMO ocean model (Madec 2011), with large scale models using free-slip and small-scale models using no-slip. A semi-slip induces the circulation expected from boundary layers but avoids under-estimation of fluid velocities in sparse grids near walls. Domain edge boundary conditions can be specified as Dirichlet conditions on velocity, or left open as free-flow.

Velocity field information from measurements, estimates or external data can either be set precisely or within a bounded range, between a minimum and maximum value determined a confidence interval. The estimated value of an uncertain state is applied prior to the projection step of flow calculation. During pressure projection the value is altered, within the bounded range, to ensure divergence free flow. If the value is at a boundary limit, then it is fixed during pressure projection and other free flow-field velocities are adjusted by the pressure field until the flow is divergence free.

For example, consider an estimate of a horizontal velocity component $\hat{u} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{N} \to \mathbb{R}$, where the horizontal velocity component is modelled as a normally distributed random variable, $u \sim N(\bar{u}, \sigma_u^2)$ where $\sigma_u^2 : \mathbb{N} \to \mathbb{R}_+$ is the variance of u. The mapping from $\mathbb{N}$

just describes this is the estimate from the kth time-step, for a particular velocity. Under an assumption of a normally distributed random variable, then a confidence interval can be described by, $\mathbb{P}[\hat{u} - k_\sigma \sigma_u \leq u \leq \hat{u} + k_\sigma \sigma_u] \approx \zeta_u$, where for each $k_\sigma \in \{1, 2, 3, 4\}$ there is a corresponding probability $\zeta_u \in \{0.6827, 0.9545, 0.9973, 0.9995\}$. In this work, a three-sigma ($k_\sigma = 3$) confidence interval is utilised unless otherwise stated. Prior to pressure projection, u is set to $\hat{u}$, then is modified in each iteration of the velocity field correction but must remain in the interval $[\hat{u} - k_\sigma \sigma_u, \hat{u} + k_\sigma \sigma_u]$.

4.1.3 Discrete Navier-Stokes solver

This section describes the discretised Navier-Stokes solver for a free-stream cell, using finite difference methods upon a regular grid spatial discretisation and a forward Euler advection and backward Euler diffusion time discretisation, then pressure projection correction to ensure divergence free flow. Although forward Euler advection is not the most accurate method, the temporal evolution of flow is dominated by the tidal dynamics. The solver is described for a representative velocity component in the horizontal and vertical directions. A free-stream cell uses centred finite difference methods, while, for example, a left hand edge cell would use single-sided finite difference instead for the left-side horizontal velocity gradient.

Let the superscript values k, k', k'', $k''a$, $k''d$, k''', k'''' be the notation of solver-steps. These are k for the initial velocity, k' for the post-application of time-varying boundaries velocity, then the advected $k''a$ or diffused $k''d$ velocity or both advected and diffused k'' velocity. Then the further superscript k''' is for the re-applied boundaries and measurements velocity, and finally the superscript k'''' is for the post-pressure-projection divergence free velocity. For example, the horizontal velocity at grid index i, j and time-step k is noted as $u^k_{i+\frac{1}{2},j}$ for $i \in [\![1, n_x]\!]$ and $j \in [\![1, n_y]\!]$, while the velocity field is noted $\vec{U}^k$. The algorithm is displayed as a flow-chart in Figure 4.3.

The first step in the Navier-Stokes solver is application of boundaries, measurements and estimates. This is described by

$$u^{k'}_{i+\frac{1}{2},j} = \text{bound}(u^k_{i+\frac{1}{2},j}, \underline{u}^k_{i+\frac{1}{2},j}, \vec{\underline{u}}^k_{i+\frac{1}{2},j}), \tag{4.2}$$

where $\underline{u}^k_{i+\frac{1}{2},j} \in \mathbb{R}$ is the Dirichlet boundary condition velocity in the horizontal direction, if

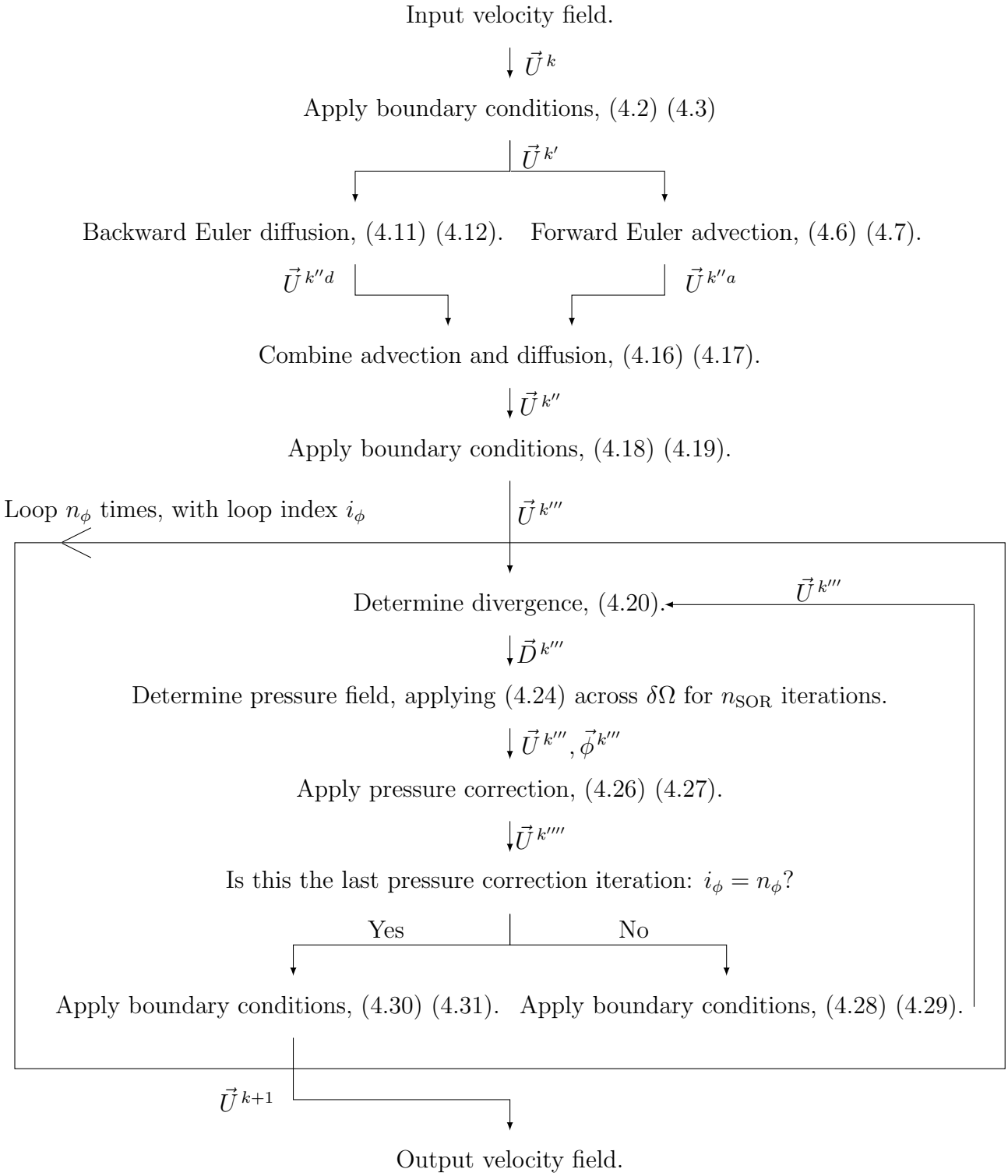


Figure 4.3: This flowchart displays the discrete Navier-Stokes solver algorithm.

present, and $\underline{\vec{u}}^k_{i+\frac{1}{2},j} \in \mathbb{R}^2$ is a monotonically increasing vector containing the lower and upper boundary values on $u^k_{i+\frac{1}{2},j}$. This could be $[\hat{u}^k_{i+\frac{1}{2},j} - k_\sigma \sigma_{u^k_{i+\frac{1}{2},j}}, \hat{u}^k_{i+\frac{1}{2},j} + k_\sigma \sigma_{u^k_{i+\frac{1}{2},j}}]$. The standard deviation $\sigma_{u^k_{i+\frac{1}{2},j}}$ is the square-root of the corresponding element in the augmented mean-squared error vector AMSE($\hat{\vec{x}}$), defined in Chapter 6 Section 6.1.1. The function, bound : $\mathbb{R} \times \mathbb{R} \times \mathbb{R}^2 \rightarrow \mathbb{R}$, either applies the Dirichlet boundary condition if there is one, or clamps the value $u^{k'}_{i+\frac{1}{2},j}$ to the interval $[\underline{\vec{u}}^k_{i+\frac{1}{2},j}(1), \underline{\vec{u}}^k_{i+\frac{1}{2},j}(2)]$, where $\underline{\vec{u}}^k_{i+\frac{1}{2},j}(1) \in \mathbb{R}$ is the lower bound and $\underline{\vec{u}}^k_{i+\frac{1}{2},j}(2) \in \mathbb{R}$ is the upper bound. These bounds are determined by external data, measurement or data assimilation confidence intervals. The application of boundaries is similar for the vertical velocity,

$$v^{k'}_{i,j+\frac{1}{2}} = \text{bound}(v^k_{i,j+\frac{1}{2}}, \underline{v}^k_{i,j+\frac{1}{2}}, \underline{\vec{v}}^k_{i,j+\frac{1}{2}}), \tag{4.3}$$

where $\underline{v}^k_{i,j+\frac{1}{2}} \in \mathbb{R}$ is the Dirichlet boundary condition velocity in the vertical direction, if present, and $\underline{\vec{v}}^k_{i,j+\frac{1}{2}} \in \mathbb{R}^2$ is a monotonically increasing vector containing the lower and upper boundary values on $v^k_{i,j+\frac{1}{2}}$.

The advection diffusion forward Euler step for the horizontal velocity is described by

$$\begin{aligned}
\frac{u^{k''}_{i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j}}{\delta t_k} = \vec{s}^k_{u,i+\frac{1}{2},j} &- u^{k'}_{i+\frac{1}{2},j} \left(\frac{-\frac{1}{2}u^{k'}_{i-\frac{1}{2},j} + \frac{1}{2}u^{k'}_{i+\frac{3}{2},j}}{\delta x} \right) \\
&- u^{k'}_{i+\frac{1}{2},j} \frac{1}{2} \left(\frac{v^{k'}_{i,j+\frac{1}{2}} - v^{k'}_{i,j-\frac{1}{2}}}{\delta y} + \frac{v^{k'}_{i+1,j+\frac{1}{2}} - v^{k'}_{i+1,j-\frac{1}{2}}}{\delta y} \right) \\
&+ \nu \left(\frac{u^{k'}_{i+\frac{3}{2},j} + u^{k'}_{i-\frac{1}{2},j} - 2u^{k'}_{i+\frac{1}{2},j}}{\delta x^2} + \frac{u^{k'}_{i+\frac{1}{2},j+1} + u^{k'}_{i+\frac{1}{2},j-1} - 2u^{k'}_{i+\frac{1}{2},j}}{\delta y^2} \right),
\end{aligned} \tag{4.4}$$

where $\vec{s}^k_{u,i+\frac{1}{2},j} \in \mathbb{R}$ is the horizontal component of the external force $\vec{s}_U$ at time-step k and grid location (i, j). The advection diffusion forward Euler step for the vertical velocity is similarly

described by

$$\frac{v^{k''}_{i,j+\frac{1}{2}} - v^{k'}_{i,j+\frac{1}{2}}}{\delta t_k} = \vec{s}^{\,k}_{v,i,j+\frac{1}{2}} - v^{k'}_{i,j+\frac{1}{2}} \left(\frac{-\frac{1}{2}v^{k'}_{i,j-\frac{1}{2}} + \frac{1}{2}v^{k'}_{i,j+\frac{3}{2}}}{\delta y} \right) - v^{k'}_{i,j+\frac{1}{2}} \frac{1}{2} \left(\frac{u^{k'}_{i+\frac{1}{2},j} - u^{k'}_{i-\frac{1}{2},j}}{\delta x} + \frac{u^{k'}_{i+\frac{1}{2},j+1} - u^{k'}_{i-\frac{1}{2},j+1}}{\delta x} \right) + \nu \left(\frac{v^{k'}_{i,j+\frac{3}{2}} + v^{k'}_{i,j-\frac{1}{2}} - 2v^{k'}_{i,j+\frac{1}{2}}}{\delta y^2} + \frac{v^{k'}_{i+1,j+\frac{1}{2}} + v^{k'}_{i-1,j+\frac{1}{2}} - 2v^{k'}_{i,j+\frac{1}{2}}}{\delta x^2} \right), \quad (4.5)$$

where $\vec{s}^{\,k}_{v,i,j+\frac{1}{2}} \in \mathbb{R}$ is the vertical component of the external force $\vec{s}_U$ at time-step k and grid location (i, j). Note that for large diffusion rates forward Euler diffusion can oscillate, then become unstable (Stam 2003), though this is unlikely here. However, since stability of the solver is critical, the backwards Euler description of diffusion is used instead, with a Successive-Over-Relaxation (SOR) solver, as in Stam 2003. In this case, (4.4) reduces to advection terms only,

$$\frac{u^{k''a}_{i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j}}{\delta t_k} = \vec{s}^{\,k}_{u,i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j} \left(\frac{-\frac{1}{2}u^{k'}_{i-\frac{1}{2},j} + \frac{1}{2}u^{k'}_{i+\frac{3}{2},j}}{\delta x} \right) - u^{k'}_{i+\frac{1}{2},j} \frac{1}{2} \left(\frac{v^{k'}_{i,j+\frac{1}{2}} - v^{k'}_{i,j-\frac{1}{2}}}{\delta y} + \frac{v^{k'}_{i+1,j+\frac{1}{2}} - v^{k'}_{i+1,j-\frac{1}{2}}}{\delta y} \right), \quad (4.6)$$

with a similar description for the vertical velocity,

$$\frac{v^{k''}_{i,j+\frac{1}{2}} - v^{k'}_{i,j+\frac{1}{2}}}{\delta t_k} = \vec{s}^{\,k}_{v,i,j+\frac{1}{2}} - v^{k'}_{i,j+\frac{1}{2}} \left(\frac{-\frac{1}{2}v^{k'}_{i,j-\frac{1}{2}} + \frac{1}{2}v^{k'}_{i,j+\frac{3}{2}}}{\delta y} \right) - v^{k'}_{i,j+\frac{1}{2}} \frac{1}{2} \left(\frac{u^{k'}_{i+\frac{1}{2},j} - u^{k'}_{i-\frac{1}{2},j}}{\delta x} + \frac{u^{k'}_{i+\frac{1}{2},j+1} - u^{k'}_{i-\frac{1}{2},j+1}}{\delta x} \right). \quad (4.7)$$

The backward Euler description of diffusion for the horizontal velocity is

$$u^{k''d}_{i+\frac{1}{2},j} - \delta t_k \nu \left(\frac{u^{k''d}_{i+\frac{3}{2},j} + u^{k''d}_{i-\frac{1}{2},j} - 2u^{k''d}_{i+\frac{1}{2},j}}{\delta x^2} + \frac{u^{k''d}_{i+\frac{1}{2},j+1} + u^{k''d}_{i+\frac{1}{2},j-1} - 2u^{k''d}_{i+\frac{1}{2},j}}{\delta y^2} \right) = u^{k'}_{i+\frac{1}{2},j}. \quad (4.8)$$

This can be rearranged to

$$\delta x^2 \delta y^2 u^{k''d}_{i+\frac{1}{2},j} - \delta y^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{3}{2},j} + u^{k''d}_{i-\frac{1}{2},j} - 2u^{k''d}_{i+\frac{1}{2},j} \right) - \delta x^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{1}{2},j+1} + u^{k''d}_{i+\frac{1}{2},j-1} - 2u^{k''d}_{i+\frac{1}{2},j} \right) = \delta x^2 \delta y^2 u^{k'}_{i+\frac{1}{2},j}, \quad (4.9)$$

and then to

$$u^{k''d}_{i+\frac{1}{2},j} + 2\delta y^2 \delta t_k \nu u^{k''d}_{i+\frac{1}{2},j} + 2\delta x^2 + \delta t_k \nu u^{k''d}_{i+\frac{1}{2},j} = u^{k'}_{i+\frac{1}{2},j} + \delta y^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{3}{2},j} + u^{k''d}_{i-\frac{1}{2},j} \right) + \delta x^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{1}{2},j+1} + u^{k''d}_{i+\frac{1}{2},j-1} \right), \quad (4.10)$$

and finally to

$$u^{k''d}_{i+\frac{1}{2},j} = \frac{\left(u^{k'}_{i+\frac{1}{2},j} + \delta y^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{3}{2},j} + u^{k''d}_{i-\frac{1}{2},j} \right) + \delta x^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{1}{2},j+1} + u^{k''d}_{i+\frac{1}{2},j-1} \right) \right)}{(1 + 2\delta y^2 \delta t_k \nu + 2\delta x^2 \delta t_k \nu)}. \quad (4.11)$$

The expression for the vertical velocity diffusion is similar,

$$v^{k''d}_{i,j+\frac{1}{2}} = \frac{\left(v^{k'}_{i,j+\frac{1}{2}} + \delta x^2 \delta t_k \nu \left(v^{k''d}_{i,j+\frac{3}{2}} + v^{k''d}_{i,j-\frac{1}{2}} \right) + \delta y^2 \delta t_k \nu \left(v^{k''d}_{i+1,j+\frac{1}{2}} + v^{k''d}_{i-1,j+\frac{1}{2}} \right) \right)}{(1 + 2\delta x^2 \delta t_k \nu + 2\delta y^2 \delta t_k \nu)}. \quad (4.12)$$

The backwards Euler diffusion equations are solved through iterative SOR. The solver first iterates (4.11) and (4.12) over the spatial domain, then repeats this spatial iteration process $n_{\text{diff}} \in \mathbb{N}$ times until the diffusion solution of $u^{k''d}_{i+\frac{1}{2},j}$ and $v^{k''d}_{i,j+\frac{1}{2}}$ converges for all i, j. In the special case where $\delta x = \delta y$, then backwards Euler diffusion equations reduce to

$$u^{k''d}_{i+\frac{1}{2},j} = \frac{u^{k'}_{i+\frac{1}{2},j} + \delta x^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{3}{2},j} + u^{k''d}_{i-\frac{1}{2},j} \right) + \delta x^2 \delta t_k \nu \left(u^{k''d}_{i+\frac{1}{2},j+1} + u^{k''d}_{i+\frac{1}{2},j-1} \right)}{1 + 4\delta x^2 \delta t_k \nu}, \quad (4.13)$$

and

$$v^{k''d}_{i,j+\frac{1}{2}} = \frac{\left(v^{k'}_{i,j+\frac{1}{2}} + \delta x^2 \delta t_k \nu \left(v^{k''d}_{i,j+\frac{3}{2}} + v^{k''d}_{i,j-\frac{1}{2}} \right) + \delta y^2 \delta t_k \nu \left(v^{k''d}_{i+1,j+\frac{1}{2}} + v^{k''d}_{i-1,j+\frac{1}{2}} \right) \right)}{(1 + 4\delta x^2 \delta t_k \nu)}. \quad (4.14)$$

The post advection and diffusion horizontal velocity is then described by

$$u^{k''}_{i+\frac{1}{2},j} = u^{k'}_{i+\frac{1}{2},j} + (u^{k''a}_{i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j}) + (u^{k''d}_{i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j}), \tag{4.15}$$

which reduces to,

$$u^{k''}_{i+\frac{1}{2},j} = u^{k''a}_{i+\frac{1}{2},j} + u^{k''d}_{i+\frac{1}{2},j} - u^{k'}_{i+\frac{1}{2},j}. \tag{4.16}$$

The post advection and diffusion vertical velocity is described by

$$v^{k''}_{i,j+\frac{1}{2}} = v^{k''a}_{i,j+\frac{1}{2}} + v^{k''d}_{i,j+\frac{1}{2}} - v^{k'}_{i,j+\frac{1}{2}}. \tag{4.17}$$

Here, (4.16) and (4.17)

These intermediate velocities after advection and diffusion need application of boundaries, described for the horizontal velocity by

$$u^{k'''}_{i+\frac{1}{2},j} = \text{bound}(u^{k''}_{i+\frac{1}{2},j}, \underline{u}^{k}_{i+\frac{1}{2},j}, \underline{\vec{u}}^{k}_{i+\frac{1}{2},j}). \tag{4.18}$$

The expression for the vertical velocity is similar,

$$v^{k'''}_{i,j+\frac{1}{2}} = \text{bound}(v^{k''}_{i,j+\frac{1}{2}}, \underline{v}^{k}_{i,j+\frac{1}{2}}, \underline{\vec{v}}^{k}_{i,j+\frac{1}{2}}). \tag{4.19}$$

Pressure projection step

The divergence, $D^{k'''}_{i,j} : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \to \mathbb{R}$ is then determined using the advected and diffused velocity field with boundaries applied. The divergence is described by

$$D^{k'''}_{i,j} = \frac{u^{k'''}_{i+\frac{1}{2},j} - u^{k'''}_{i-\frac{1}{2},j}}{\delta x} + \frac{v^{k'''}_{i,j+\frac{1}{2}} - v^{k'''}_{i,j-\frac{1}{2}}}{\delta y}. \tag{4.20}$$

Let $\vec{D}^{\,k'''} : \mathbb{N} \to \mathbb{R}^{n_x n_y}$ be the vector of $D^{k'''}_{i,j}$ across the discretised spatial domain, and $\vec{\phi}^{\,k'''} : \mathbb{N} \to \mathbb{R}^{n_x n_y}$ be the vector of pressure p across the domain for the velocities $\vec{U}^{\,k'''} \in \mathbb{R}^{2n_x n_y}$. The pressure Poisson equation is then described by

$$\nabla^2 \vec{\phi}^{\,k'''} = \vec{D}^{\,k'''}. \tag{4.21}$$

The discrete form of (4.21) using a 3-point centred difference is

$$\frac{p^{k'''}_{i+1,j} + p^{k'''}_{i-1,j} - 2p^{k'''}_{i,j}}{\delta x^2} + \frac{p^{k'''}_{i,j+1} + p^{k'''}_{i,j-1} - 2p^{k'''}_{i,j}}{\delta y^2} = D^{k'''}_{i,j}. \tag{4.22}$$

This is solved with SOR, an iterative method. Let (4.22) be rearranged to

$$\left(2\delta y^2 + 2\delta x^2\right) p^{k'''}_{i,j} = -\delta x^2 \delta y^2 D^{k'''}_{i,j} + \delta y^2 \left(p^{k'''}_{i+1,j} + p^{k'''}_{i-1,j}\right) + \delta x^2 \left(p^{k'''}_{i,j+1} + p^{k'''}_{i,j-1}\right), \tag{4.23}$$

and then to

$$p^{k'''}_{i,j} = \frac{1}{(2\delta y^2 + 2\delta x^2)} \left(-\delta x^2 \delta y^2 D^{k'''}_{i,j} + \delta y^2 \left(p^{k'''}_{i+1,j} + p^{k'''}_{i-1,j}\right) + \delta x^2 \left(p^{k'''}_{i,j+1} + p^{k'''}_{i,j-1}\right)\right). \tag{4.24}$$

In the special case where $\delta x = \delta y$, this reduces to

$$p^{k'''}_{i,j} = \frac{1}{4} \left(-\delta x^2 D^{k'''}_{i,j} + \left(p^{k'''}_{i+1,j} + p^{k'''}_{i-1,j} + p^{k'''}_{i,j+1} + p^{k'''}_{i,j-1}\right)\right). \tag{4.25}$$

The pressure is evaluated across the spatial domain for all $i \in [\![1, n_x]\!]$ and $j \in [\![1, n_y]\!]$, using (4.24). This is then repeated $n_{\text{SOR}} \in \mathbb{N}$ times, or until convergence. For simplicity and speed of evaluation, $n_{\text{SOR}} = 20$ is used here, which is too low to converge to a high degree of accuracy, but this is accounted for as the pressure projection step is repeated a further n_ϕ times. With the pressure field determined, the velocity field is then modified to be divergence free by

$$u^{k''''}_{i+\frac{1}{2},j} = u^{k'''}_{i+\frac{1}{2},j} + \frac{p^{k'''}_{i+1,j} - p^{k'''}_{i,j}}{\delta x}, \tag{4.26}$$

and

$$v^{k''''}_{i,j+\frac{1}{2}} = v^{k'''}_{i,j+\frac{1}{2}} + \frac{p^{k'''}_{i,j+1} - p^{k'''}_{i,j}}{\delta y}. \tag{4.27}$$

Finally, the boundary values are applied again. If this pressure projection iteration $i_\phi \in [\![1, n_\phi]\!]$ fulfils $i_\phi < n_\phi$, then boundary application is described by

$$u^{k'''}_{i+\frac{1}{2},j} = \text{bound}(u^{k''''}_{i+\frac{1}{2},j}, \underline{u}^{k}_{i+\frac{1}{2},j}, \underline{\vec{u}}^{k}_{i+\frac{1}{2},j}), \tag{4.28}$$

and

$$v^{k'''}_{i,j+\frac{1}{2}} = \text{bound}(v^{k''''}_{i,j+\frac{1}{2}}, \underline{v}^{k}_{i,j+\frac{1}{2}}, \underline{\vec{v}}^{k}_{i,j+\frac{1}{2}}). \tag{4.29}$$

The Navier-Stokes solver returns to (4.20). Equations (4.20), (4.24), (4.26), (4.27), (4.30) and (4.31) are repeated to produce divergence free flow, with boundary values, estimates and measurements incorporated. This is displayed in Figure 4.3. Note that the repeated correction through pressure projection propogates pressure through the domain, unlike a single pressure correction step. Hence, this can be thought of as a weakly compressible system.

Otherwise, if $i_\phi = n_\phi$ then the velocity field at the next time-step $k+1$ is determined,

$$u^{k+1}_{i+\frac{1}{2},j} = \text{bound}(u^{k''''}_{i+\frac{1}{2},j}, \underline{u}^{k}_{i+\frac{1}{2},j}, \underline{\vec{u}}^{k}_{i+\frac{1}{2},j}), \tag{4.30}$$

and

$$v^{k+1}_{i,j+\frac{1}{2}} = \text{bound}(v^{k''''}_{i,j+\frac{1}{2}}, \underline{v}^{k}_{i,j+\frac{1}{2}}, \underline{\vec{v}}^{k}_{i,j+\frac{1}{2}}). \tag{4.31}$$

4.1.4 Wind flow

Calculation of wind velocity $\vec{U}_\text{w}$ is handled by the 2D Navier-Stokes solver described in Section 4.1.2. Though the grid is shared, the time-step requirement of the wind solver is likely to be a much smaller time-step due to the higher velocities of wind flow. Hence, the wind velocity solver steps forward in multiple smaller, sub-time-steps, to form δt, and only the final solution at the shared time $t + \delta t$ is utilised in the state trajectory. In the wind solver there is replacement of zero flow boundaries for obstacles by maximum wind-speed conditions to represent wind resistant areas such as cities or mountains. This acts as a flow restriction and thus resolves external data forced flow to greater accuracy for local geographic features. Velocity limits are imposed in the same manner as estimated values in Section 4.1.2.

The velocity limit $\vec{U}_{\text{w}_\text{lim}} \coloneqq \vec{U}_{\text{w}_\text{lim}}(x, y) : \partial\Omega \to \mathbb{R}^2$ is calculated by the urban canopy profile

$$\vec{U}_{\text{w}_\text{lim}} = (1 - \lambda_p)^2 \left\| \vec{U}_{\text{w}_\text{max}} \right\|_2, \tag{4.32}$$

where $\lambda_p \coloneqq \lambda_p(x, y) : \partial\Omega \to [0, 1]$ is the obstruction plan, or footprint, density in the cell area at 10 m altitude (CERC 2017). If an obstruction density map is not available, a coefficient can

be specified in place of $(1-\lambda_p)^2$ for coastal and land cells empirically using local data.

It is assumed that the external wind data incorporates the effect of land topology (height) on wind flow, justified by the free-availability of data from a multitude of global circulation wind models. If there is no alternative, the spatial derivatives in the horizontal and vertical directions of the land height-map can be utilised to estimate an obstruction density map, under an assumption that 2D wind flow is always parallel to the ground. Note this is a tenuous method that operates as a last resort should no other data or model be available for generation of wind-flow data that accounts for land topology.

For an intuitive example: Upon a discretised grid of spacing δx and δy, there is flow from a cell with height 0 m into a cell of height 5 m. Assuming the down-wind cell resembles a 5 m high step, the volume of obstruction in the down-wind cell is $5\delta x\delta y$. Recall that here, wind is simulated at a height of 10 m and hence the obstruction volume is modified to now stand at 10 m high. Therefore, the floor-plan area obstructed is reduced to $\frac{5}{10}\delta x\delta y$, or simply $\frac{1}{2}$ as a proportion of the cell area. However, this obstruction is only present if the wind flow is up-hill and if the slope is steepening. Furthermore a topology based obstruction should never present a complete flow blockage, so 0.95 is selected as an upper-limit on an obstruction coefficient.

The obstruction in the horizontal direction, $\lambda_{p_x} : \partial\Omega \to [0, 0.95]$, is described by

$$\lambda_{p_x} = \text{median}\left(0, \frac{1}{10}\text{sgn}\left(u_\text{w}\right)\text{sgn}\left(\nabla_x^2\partial\Omega_H\right)\nabla_x\partial\Omega_H, 0.95\right), \tag{4.33}$$

with a similar expression for the vertical direction. In (4.33), the median function clamps λ_{p_x} between 0 and 1, $\partial\Omega_H : \partial\Omega \to \mathbb{R}$ is the height map of the surface $\partial\Omega$ and the operation $\nabla_x\partial\Omega_H$ returns the gradient of the height-map in the horizontal direction, using single-sided finite difference with the up-wind cell. The operation $\nabla_x^2\partial\Omega_H$ returns the second derivative of the height-map in the horizontal direction. The term $\text{sgn}\left(u_\text{w}\right)$ ensures an obstruction only exists if flow is moving up-hill and $\text{sgn}\left(\nabla_x\partial\Omega_H\right)$ ensures an obstruction exists only if the slope is steepening, otherwise the wind is assumed to already be flowing parallel to the slope. In Cindori et al. 2020, it is extrapolated from West and Apelt 1982 that the effect of an obstruction less than 6% is negligible and we account for this with a modified obstruction

variable $\bar{\lambda}_{p_x} : \partial\Omega \to [0, 0.95]$, described by

$$\bar{\lambda}_{p_x} = \begin{cases} 0 & \lambda_{p_x} < 0.06 \\ \lambda_{p_x} & \text{otherwise,} \end{cases} \tag{4.34}$$

again with a similar expression for the vertical direction. Note that unlike (4.32), the topology based obstruction operates on a velocity component basis, not a velocity vector basis. So for the horizontal component, the limit is described by,

$$u_{\text{Wlim}} = (1 - \lambda_p)^2 u_{\text{Wmax}}, \tag{4.35}$$

with a similar expression for the vertical component.

4.1.5 Ocean flow

The Navier-Stokes equation are solved in 2D, but a depth velocity profile extension to 2.5D is important when modelling oil-trajectories and the separation of slicks caused by sub-surface shear flows and entrainment. Note that this velocity profile approach acceptable for oil surface slick simulation, but for an oil plume, or up-welling from a sunken vessel, a full 3D ocean simulation is likely required. Typical individual velocity profiles for the 2.5D approach are shown in Figure 4.4.

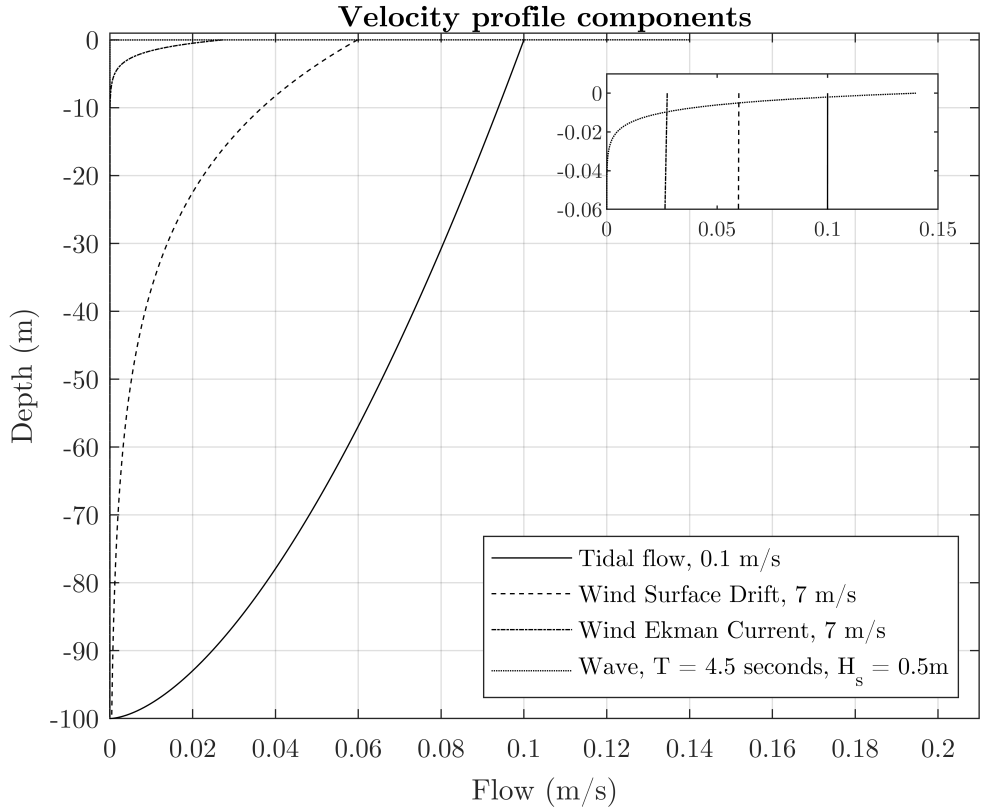


Figure 4.4: A depiction of sub-surface flow resulting from mechanisms included in the three-dimensional model. The insert magnifies the sub-surface flow at shallow depths, note the very shallow effect of wind surface drift. Here, a depth of -5 m is 5 m below the water surface.

Tidal and circulation flow

The velocity profile, introducing vertical variation to $\vec{U}_{\mathrm{c}}$ for a tide driven flow, follows a standard logarithmic profile where $z \in [0, \bar{z}] \subset \mathbb{R}$ is depth in the water column, with $z = 0$ at the surface and $z = \bar{z}$ at maximum water depth where $\bar{z}$ is the mean total water depth in that cell. A no-slip condition, $\vec{U}_{\mathrm{c}_{\bar{z}}} = 0$ is imposed on the sea floor and $\vec{U}_{\mathrm{c}_z}$ increases to its maximum value at the surface. As predictions focus on surface oil particles, boundary layer simulation is omitted and U_{c_z} can be simply described by

$$\vec{U}_{\mathrm{c}_z} = \vec{U}_{\mathrm{c}_0} \left(1 - \frac{z}{\bar{z}}\right)^{\frac{1}{6}}, \tag{4.36}$$

where the empirical denominator parameter in the power law has been assigned the value of 6, which falls in the range of accepted values for ebbing and flowing tides (Thiébaut and Sentchev 2016).

Wind induced surface shear

Under strong wind conditions the velocity of surface water is heavily affected by the boundary stress between the two-phase flow of air and water, so is vital for inclusion in an advective ocean model. Large scale models often use measured wind speed data (De Dominicis et al. 2013a) or wind speed estimated from surface roughness (measured via radar scattering) (Smith 1988) to simply calculate a surface flow velocity. This takes the form of a scaled velocity $\alpha_{\mathrm{w}} \left\| \vec{U}_{\mathrm{w}} \right\|_2$, rotated by a wind drift angle $\beta \coloneqq \beta(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$, representing the balancing of wind shear and Coriolis effects. This velocity is then scaled to a logarithmic velocity profile (Wu 1975), modelled in oceans as beginning at z_0 (the wind driven surface layer) and falling to zero effect at z_c meters (Proctor, Flather, and Elliott 1994). This latter depth can be approximated as

$$z_c \approx \alpha_z L. \tag{4.37}$$

A value of $\alpha_z = 2$ is suggested to give good agreement with observations (Elliott 1986) in a short-fetch environment, using L as the dominant wavelength of sea-surface waves. For even a low wind speed fetch in deep water, wave lengths are likely to be around 8 meters giving rise to large z_c values and a large effect of wind on sub-surface currents, with data supporting a wind penetration depth of 40 meters (Elliott 1986).

An assumption of the above method is instantaneous changes of sub-surface currents in response to local wind gusting. Here, wind effects are modelled in parts, deep effects are modelled as a combination of slow time-varying Ekman currents and stokes drift from a linear wave model. Shallow effects are instantaneously applied by a logarithmic velocity profile. In a wind wave spectrum, local wind affects only the small-scale ripples (capillary waves) and gravity-wind waves that are accounted for in the linear wave model. Modelling just capillary waves gives rise to varying surface roughness (as wave amplitude) across wind conditions, of typical wavelength (Lamb 1895) defined by

$$L_{\mathrm{capillary}} = 2\pi \sqrt{\frac{\sigma_{\mathrm{water}}}{(\rho_{\mathrm{water}} - \rho_{\mathrm{air}})g}}, \tag{4.38}$$

where for an air-water interface, $L_{\mathrm{capillary}} = 0.017\mathrm{m}$ (Lamb 1895) and σ_{water} is the surface tension of water, ρ_{water} and ρ_{air} are the densities of water and air, respectively. Thus the new wind shear zero effect depth for $\alpha_z = 2$ is 0.037m when using (4.38) to determine the wavelength in

4.37. This is a shallow depth, where viscous shear and vertical mixing allows an assumption of a velocity change time-scale much smaller than the simulation time-step. Hence velocity changes immediately with fluctuating wind as in traditional percentage based algorithms for surface oil spill drift due to wind/wave interaction (Spaulding 2017), with the wind shear velocity profile $\vec{U}_{\mathrm{w}_z} \coloneqq \vec{U}_{\mathrm{w}_z}(x, y, z, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^2$, defined by

$$\vec{U}_{\mathrm{w}_z} = \alpha_{\mathrm{w}} \vec{U}_{\mathrm{w}} e^{-\frac{2\pi}{z_c} z}, \tag{4.39}$$

where $\alpha_{\mathrm{w}} \in [0.005, 0.03]$. A value of 0.02 is suggested for α_{w} (Proctor, Flather, and Elliott 1994), but varies within literature (Kim et al. 2014).

Ekman currents

Ekman currents describe the net motion of fluid that results from the balance of a forcing wind, turbulent drag and Coriolis forces. In a small scale simulation it would be preferable not to assume an instantaneous (in distance and time) change in the sub-surface layer velocity due to wind. Ekman currents take approximately 12 hours to form ($T_{\mathrm{E}} = 12$ hours), accelerating approximately linearly to their fully formed magnitude (Weatherly 1975). Ideally, the Ekman current would change towards its final value at each time-step, but this would require changing every depth value in every grid cell, at every time-step, leading to excessive computational load. An alternative would be to keep a moving average of the last 12 hours of wind data, but this requires stored data and introduces a large phase lag in Ekman changes. For a domain where wind-speed changes are frequent, an incremental weighted mean of wind speeds to form an average of the past 12 hours of wind speed is proposed. The Ekman wind velocity $\vec{U}_{\mathrm{w_E}} \coloneqq \vec{U}_{\mathrm{w_E}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^2$, is calculated for the surface of each grid cell:

$$\vec{U}_{\mathrm{w_E}} = \frac{W_{\mathrm{E}_1} \vec{U}^{t-\delta t}_{\mathrm{w_E}} + W_{\mathrm{E}_2} \vec{U}^{t}_{\mathrm{w}}}{W_{\mathrm{E}_1} + W_{\mathrm{E}_2}}, \tag{4.40}$$

where the Ekman averaged wind velocity at the previous time-step is $\vec{U}^{t-\delta t}_{\mathrm{w_E}}$ and $\vec{U}^{t}_{\mathrm{w}}$ is the current time wind-velocity. The weights for the value $W_{\mathrm{E}_1} \in \mathbb{R} : \ [0, 1]$ and future values $W_{\mathrm{E}_2} \in \mathbb{R} : \ [0, 1]$ are calculated as

$$W_{\mathrm{E}_1} = \frac{T_{\mathrm{E}} - \delta t}{\frac{1}{2} T_{\mathrm{E}}} \tag{4.41a}$$

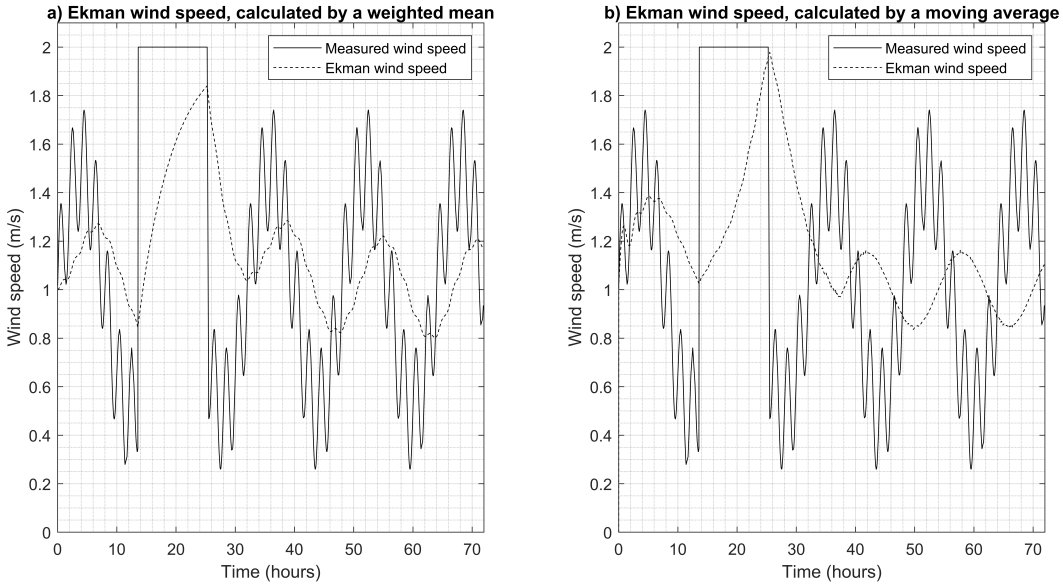


Figure 4.5: A depiction of the wind speed $\vec{U}_{\mathrm{w}}$ that Ekman currents are calculated from, $\vec{U}_{\mathrm{wE}}$, under noisy wind conditions. a) Calculated using the weighted mean approach. b) Calculated by a traditional 12 hour moving average approach, which shows a more linear growth but significantly greater lag. Observe that at 38 hours, the wind speed falls beneath the Ekman wind speed. Note how the weighted mean approach to calculating Ekman wind speed (left) immediately begins to reduce, while the moving average approach (right) has 4 hours of lag before it decreases.

and

$$W_{\mathrm{E}_2} = \frac{\delta t}{T_{\mathrm{E}}}. \tag{4.41b}$$

Figure 4.5 shows the growth and decay of the Ekman wind speed, used to calculate the Ekman current, under a range of wind conditions when calculated by both the weighted mean and moving average approaches. Results from literature suggest there should be no lag between wind stress and Ekman shelf velocities (Kirincich and Barth 2008). Inspection of Figure 4.5 clearly shows the lag from the moving average approach is significantly greater than that from the weighted mean approach, to the point where the Ekman wind speed response is almost completely out of phase with the forcing wind. The wind stress of the surface layer can be used to estimate the Ekman current magnitude. Prior work provides stress coefficients for water under a variety of conditions, including adjustment factors for wind speeds measured at various heights to normalise their values at 10 m above sea level (Wu 1980) and (Smith 1988). Let the stress coefficient $C_{D_{\mathrm{stress}}} \coloneqq C_{D_{\mathrm{stress}}}(x, y, t, \left\|\vec{U}_{\mathrm{wE}}\right\|_2) : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}_+$. Hence, using $\vec{U}_{\mathrm{wE}}$ as wind velocity yields

$$\tau = C_{D_{\mathrm{stress}}} \rho_{\mathrm{air}} \left\|\vec{U}_{\mathrm{wE}}\right\|_2^2, \tag{4.42}$$

where the scalars $\tau \coloneqq \tau(x, y, t) : \partial\Omega \times \mathbb{R}_+ \in \mathbb{R}_+$ are the wind stresses on the water surface and $\rho_{\text{air}} \in \mathbb{R}_+$ is the air density. A final Ekman current velocity profile is calculated (Pond and Pickard 1983), using a vertical eddy viscosity coefficient (Rasmussen 1985) of $A_z \coloneqq A_z(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ and a surface Ekman speed $V_{0_{\text{E}}} \coloneqq V_{0_{\text{E}}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ defined by

$$A_z = 4.3 \times 10^{-4} \left\| \vec{U}_{\text{wE}} \right\|_2^2 \tag{4.43a}$$

and

$$V_{0_{\text{E}}} = \frac{\sqrt{2}\pi\tau}{z_{\text{E}}\rho_{\text{water}}|f|}, \tag{4.43b}$$

where $z_{\text{E}} \coloneqq z_{\text{E}}(x, y, z, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ is the Ekman layer depth (Pond and Pickard 1983), $f \coloneqq f(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ is the Coriolis frequency and $\rho_{\text{water}} \in \mathbb{R}_+$ is the water density. Adjusting for a coordinate system where u is positive east velocity and v is positive north, with an ascending z with depth and positive clockwise from north angles, an alternative formulation that also reflects the smaller drift divergence angle in current formations under high wind conditions can be described by

$$u_{\text{E}_z} = \pm V_{0_{\text{E}}} \sin\left(\beta_{\text{rad}} - \frac{\pi}{z_{\text{E}}} z\right) e^{-\frac{\pi}{z_{\text{E}}} z}, \tag{4.44a}$$

where the negative sign applies to the northern hemisphere, the positive to the southern hemisphere. Similarly

$$v_{\text{E}_z} = \pm V_{0_{\text{E}}} \cos\left(\beta_{\text{rad}} - \frac{\pi}{z_{\text{E}}} z\right) e^{-\frac{\pi}{z_{\text{E}}} z}, \tag{4.44b}$$

where a wind drift angle (Wang and Shen 2010) is proposed for the Ekman current angle, instead of a constant 45 degrees:

$$\beta = \begin{cases} 40^\circ - 8^\circ \sqrt[4]{u_{\text{w}}^2 + v_{\text{w}}^2} & \text{for } 0 \leq \sqrt{u_{\text{w}}^2 + v_{\text{w}}^2} \leq 25\,\text{m/s}, \\ 0^\circ & \text{for } \sqrt{u_{\text{w}}^2 + v_{\text{w}}^2} > 25\,\text{m/s}, \end{cases} \tag{4.45a}$$

and

$$\beta_{\text{rad}} = \beta \frac{\pi}{180}. \tag{4.45b}$$

The velocity components (4.44) compose the fully formed Ekman velocity $\vec{U}_{c_{E_z}} = [u_{E_z}, v_{E_z}]^T$, where $\vec{U}_{c_{E_z}} \coloneqq \vec{U}_{c_{E_z}}(x, y, z, t) : \Omega \times \mathbb{R}_+ \to \mathbb{R}^2$.

These equations produce an Ekman velocity profile, shown in Figure 4.6, that follows a typical spiral pattern and has a magnitude of approximately 1% of the wind speed. This is as expected, the 3% wind velocity advection employed by classical models will be a composite of the smaller Ekman currents, Stokes drift and Surface stress induced currents calculated separately here.

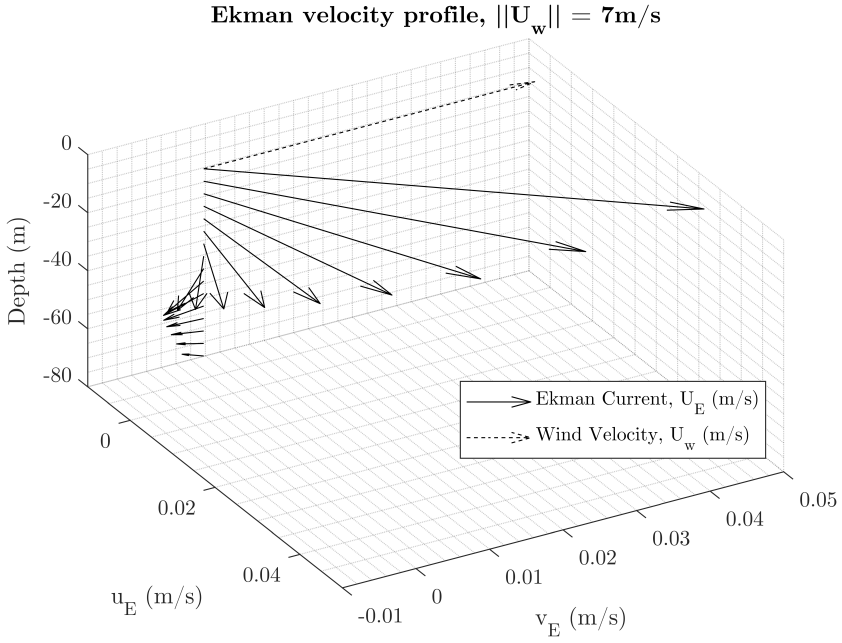


Figure 4.6: The spatial variation of Ekman velocity with depth resulting from non-aligned wind and current angles. Here, a depth of -5 m is 5 m below the water surface.

4.1.6 Linear wave model

To determine the effect of waves on contaminants a spatio-temporally varying wave spectrum is approximated by the Sheffield Wave Environment Model (SWEM) (Heins and Jones 2016), which combines modified wave spectra from ocean swell, local wind, surface current and finite water depth to simulate the ocean surface. It includes a directional spreading function and swell estimation from fetch parameters or buoy data. Each cell updates its wave model with the local wind and surface current velocity at every time-step. The wave models then re-evaluate the wave spectra, along with the significant wave height $H_s \coloneqq H_s(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$, wavelength $L \coloneqq L(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ and wave period $T \coloneqq T(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ for each grid cell. In a time-constrained simulation, the wave model is updated only in cells where oil is present without adversely affecting results.

4.2 Oil model

The oil model uses a common Lagrangian approach (Spaulding 2017), utilising large numbers of particles (see section 4.2.5), each representing a volume of contaminant. Particles undergo advection and turbulent diffusion in response to forcing from the environmental model. Particles are then used to build a thickness map and undergo mechanical spreading in areas where the thickness is above a minimum value, with particle size determined from oil properties. Particles can be entrained underwater, determined by variables from the wave model, with subsequent resurfacing dependent upon vertical turbulent diffusion and terminal buoyancy velocity.

4.2.1 Advection and diffusion

The advective velocity of particles at depth z consists of horizontal velocity components u_{o_z}, v_{o_z} and a vertical velocity component w_{o_z}. These are determined from a summation of tidal, wind induced surface shear and Ekman current velocities, plus turbulent diffusion terms as follows:

$$\begin{bmatrix} u_{\mathrm{o}_z} \\ v_{\mathrm{o}_z} \\ w_{\mathrm{o}_z} \end{bmatrix} = \alpha_{\mathrm{w_o}} \begin{bmatrix} u_{\mathrm{w}} \\ v_{\mathrm{w}} \\ 0 \end{bmatrix} + \alpha_{\mathrm{c_o}} \begin{bmatrix} u_{c_z} \\ v_{c_z} \\ 0 \end{bmatrix} + \begin{bmatrix} u_{\mathrm{w}_z} \\ v_{\mathrm{w}_z} \\ 0 \end{bmatrix} + \begin{bmatrix} u_{E_z} \\ v_{E_z} \\ 0 \end{bmatrix} + \begin{bmatrix} u_{\mathrm{wave}_z} \\ v_{\mathrm{wave}_z} \\ 0 \end{bmatrix} + \begin{bmatrix} u_d \\ v_d \\ w_b \end{bmatrix} + \begin{bmatrix} u' \\ v' \\ w'_z \end{bmatrix}, \tag{4.46}$$

where $\alpha_{\mathrm{w_o}} \in [0, 0.05]$ is a coefficient for additional wind advection and $\alpha_{\mathrm{c_o}} \in [0.95, 1.1]$ is an advection coefficient for tidal currents. The buoyancy velocity is $w_b \coloneqq w_b(x, y, z, t, D_v) : \Omega \times \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$, while diffusion correction velocities $u_d \coloneqq u_d(x, y, t, D_h) : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$ and $v_d \coloneqq v_d(x, y, t, D_h) : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R} \to \mathbb{R}$ are defined

$$u_d(x, y, t, D_h) = \frac{\delta D_h}{\delta x}, \tag{4.47a}$$

in the horizontal x direction and

$$v_d(x, y, t, D_h) = \frac{\partial D_h}{\partial y} \tag{4.47b}$$

in the horizontal y direction. They are the spatial derivative of $D_h \coloneqq D_h(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$, the horizontal diffusion coefficient (Hunter, Craig, and Phillips 1993). The turbulent diffusion

velocities comprise of $u' \coloneqq u'(x,y,t) : \partial\Omega \times \mathbb{R}_+ \in \mathbb{R}$, $v' \coloneqq v'(x,y,t) : \partial\Omega \times \mathbb{R}_+ \in \mathbb{R}$, $w' \coloneqq w'(x,y,z,t) : \Omega \times \mathbb{R}_+ \in \mathbb{R}$ in the horizontal x and y plane and vertical z direction respectively. The stokes drift velocities (defined in Section 4.2.1) $u_{\text{wave}_z} \coloneqq u_{\text{wave}_z}(x,y,z,t) : \Omega \times \mathbb{R}_+ \in \mathbb{R}$, $v_{\text{wave}_z} \coloneqq v_{\text{wave}_z}(x,y,z,t) : \Omega \times \mathbb{R}_+ \in \mathbb{R}$ in the x and y horizontal direction respectively. The additional wind advection represents only the carrying of oil droplets by wind, since the major wind drift is accounted for in the hydrodynamic model.

Turbulent diffusion is calculated by the common random walk method (Spaulding 2017), but avoids direct parameter setting for horizontal diffusivity and vertical diffusivity coefficients in favour of empirical formulae that also introduce variation in the diffusion coefficient dependent upon flow properties. Spatial variation in diffusion coefficient results in a requirement for a diffusion correction velocity (Hunter, Craig, and Phillips 1993). Horizontal turbulent diffusion velocity is assumed constant with depth and calculated (Chao, Shankar, and Wang 2003) using:

$$u' = \xi_p \sqrt{\frac{12 D_h}{\delta t}} \sin(2\pi\phi_p), \tag{4.48a}$$

$$v' = \xi_p \sqrt{\frac{12 D_h}{dt}} \cos(2\pi\phi_p), \tag{4.48b}$$

where $\xi_p \in [0,1]$ and $\phi_p \in [0,1]$ are particle specific random variables with uniform distribution. Vertical turbulent diffusion velocity is depth dependent and calculated (Lardner and Gunay 2000) according to:

$$w' = (2\zeta_w - 1)\sqrt{\frac{6 D_{v_z}}{dt}}, \tag{4.49}$$

where ζ_w is a particle specific random variable with uniform distribution in $[0,1]$. Coefficients for horizontal diffusivity D_h (Baldauf and Zängl 2012) and vertical diffusivity $D_{v_z} \coloneqq D_{v_z}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ (Ichiye 1967) are calculated as follows:

$$D_h = \frac{c_{\text{smag}}}{\frac{1}{\partial x^2 + \partial y^2}} \sqrt{T_{\text{smag}} + S_{\text{smag}}}, \tag{4.50a}$$

$$T_{\text{smag}} = \frac{\partial u_{\text{c}}}{\partial x} - \frac{\partial v_{\text{c}}}{\partial y}, \tag{4.50b}$$

$$S_{\text{smag}} = \frac{\partial u_{\text{c}}}{\partial y} + \frac{\partial v_{\text{c}}}{\partial x}, \tag{4.50c}$$

$$D_{v_z} = 0.028 \frac{H_s^2}{T} e^{-2\frac{1}{L}z} - 0.056 \frac{H_s^2}{LT} e^{-2\frac{1}{L}z}, \tag{4.50d}$$

where $c_{\text{smag}} \in \mathbb{R} : [0.01, 0.3]$ is an empirical coefficient, with a nominal default value of 0.1. Note that (4.50d) is an implementation of Ichiye 1967 corrected according to the derivative, as in Hunter, Craig, and Phillips 1993 and Nordam et al. 2019b. Further define $d_h(u, v) := D_h(x, y, t) : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, as an equivalent term to the horizontal diffusivity coefficient D_h that explicitly describes the dependencies of D_h as dependencies on velocity, u and v.

Stokes Drift

Stokes drift is the net horizontal movement of a particle due to wave motion, resulting from shear stresses and mixing layers from surface gravity waves. For each grid cell in which there are oil particles and for each time-step, the spectral wave model SWEM is used to compute the wave parameters that govern Stokes drift, chiefly significant wave height H_{s}, wavelength L and wave period T. These are evaluated from the peak magnitude $a_{\text{p}} := a_{\text{p}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ and corresponding peak frequency $f_{\text{p}} := f_{\text{p}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ of the wave spectrum.
Webb proposes the use of the peak frequency, with a Stokes drift amplitude modified by the spectral moment (calculated through intergrands) and empirical terms specific to that spectrum (Webb and Fox-Kemper 2011). The spectrum in SWEM is a summation of several others and therefore this approach would require multiple calculations of spectral moments and ultimately, too much computation. Hence, only the peak information of the SWEM spectrum (representing the fetch, local current and local wind interaction) is used, as the high frequency ripple waves are accounted for through wind shear.
Stokes drift magnitude is similar to a near-surface tidal shear (Elliott 1986) or 1 - 2% of the wind speed (Proctor, Flather, and Elliott 1994). The literature suggested (Elliott 1986) hyperbolic trigonometric formulation of Stokes drift can become undefined in deep water conditions, hence

it is redefined to give the Stokes drift speed:

$$\left\|\vec{U}_{\text{wave}_z}\right\|_2 = \omega k a_p^2 e^{-2kz}, \tag{4.51}$$

where $\omega = 2\pi/T_{\text{peak}}$, $k = 2\pi/L_{\text{peak}}$ using the wave spectrum peak values from SWEM. To achieve an accurate Stokes drift velocity, the wave spectrum produces an average wave energy direction and scales the Stokes drift velocity to the proportion of wave energy in that direction compared to the total wave energy in the spectrum. The direction and magnitudes of the waves are expressed in polar coordinates as follows:

$$\theta_{\Psi_{\text{Total}_i}} = \text{atan2}\left(\frac{k_y}{k_x}\right), \tag{4.52a}$$

$$r_{\Psi_{\text{Total}_i}} = \Psi_{\text{Total}}(k_x, k_y), \tag{4.52b}$$

where $\Psi_{\text{Total}}(k_x, k_y) \coloneqq \Psi_{\text{Total}}(k_x(x,y,t), k_y(x,y,t)) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ is the energy of the waves with wavenumbers $k_x \coloneqq k_x(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ and $k_y \coloneqq k_y(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$. The polar angle $\theta_{\Psi_{\text{Total}_i}} \coloneqq \theta_{\Psi_{\text{Total}_i}}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ and magnitude $r_{\Psi_{\text{Total}_i}} \coloneqq r_{\Psi_{\text{Total}_i}}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ form the polar coordinate representation of that wave-number, with magnitude being the wave energy and angle as the wave direction. The wave spectrum is thus converted from 2D $[k_x, k_y]$ wave numbers to a $k_x k_y$ by 1 vector of polar coordinates. The sum of the vector of polar coordinates provides an average wave energy polar coordinate with magnitude and direction of the average wave energy:

$$[\theta_{\text{sum}}, r_{\text{sum}}] = \sum_{i=1}^{k_x k_y} [\theta_{\Psi_{\text{Total}_i}}, r_{\Psi_{\text{Total}_i}}], \tag{4.53a}$$

$$\Psi_{\text{avg}_\theta} = r_{\text{sum}}. \tag{4.53b}$$

The polar angle $\theta_{\text{sum}} \coloneqq \theta_{\text{sum}}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ and magnitude $r_{\text{sum}} \coloneqq r_{\text{sum}}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ form the polar coordinate with magnitude and direction equivalent to the average wave energy. This wave energy magnitude $\Psi_{\text{avg}_\theta} \coloneqq \Psi_{\text{avg}_\theta}(x,y,t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ is used to attenuate stokes drift velocity by the fraction of wave energy that is in the average

wave direction $\Psi_{fr} \coloneqq \Psi_{fr}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$, calculated by

$$\Psi_{fr} = \frac{\Psi_{\text{avg}_\theta}}{\sum_{i=1}^{k_x k_y} \Psi_{\text{Total}_i}(k_x, k_y)}. \tag{4.54}$$

Stokes drift speed $\left\|\vec{U}_{\text{wave}_z}\right\|_2$ is in the direction of Ψ_{avg_θ}, where $\vec{U}_{\text{wave}_z} : \Omega \times \mathbb{R}_+ \to \mathbb{R}^3$ is the stokes drift velocity vector $\vec{U}_{\text{wave}_z} = [u_{s_z}, v_{s_z}, 0]^T$, forming a stokes drift velocity:

$$\vec{U}_{\text{wave}_z} = \omega k a_p^2 e^{-2kz} \Psi_{fr}. \tag{4.55}$$

4.2.2 Entrainment and buoyancy

Oil entrainment from the surface slick to the water column represents the movement of oil particles underwater by wave action and can be modelled as a random process with a probability for a particle to be entrained at a given time. The principal variable in the volume of oil entrained is the rate-scale scalar $\lambda_{\text{ow}} \coloneqq \lambda_{\text{ow}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ (Tkalich and Chan 2002), which is defined by

$$\lambda_{\text{ow}} = \frac{\pi k_e \gamma H_s}{8\alpha T_{\text{peak}} L_{\text{ow}}}, \tag{4.56}$$

where $k_e \in [0.3, 0.5] \subset \mathbb{R}$ is an empirical constant, H_s is the peak significant wave height, T_{peak} is the wave period from the linear wave model and $L_{\text{ow}} \in \mathbb{R}_+$ is a vertical length scale parameter that depends on the type of breaking wave. This is valued between 10 m and 20 m (Tkalich and Chan 2002). The vertical mixing term coefficient is $\alpha \in \mathbb{R} \ : \ [1.15, 1.85]$. The parameter $\gamma \coloneqq \gamma(x, y, t) : \partial\Omega \times \mathbb{R}_+ \in \mathbb{R}$ is a dimensionless damping coefficient that takes the following values:

$$\gamma = \begin{cases} 10^5 \omega E_{\text{w}}^{0.25}, \text{ for white-capping waves,} \\ 1.8 \times 10^{-7} \omega^3, \text{ for swell decay,} \end{cases}$$

where $E_{\text{w}} \coloneqq E_{\text{w}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ is calculated by

$$E_{\text{w}} = \frac{g \rho_{\text{water}} H_s^2}{16}, \tag{4.57}$$

where g is the gravitation acceleration constant of 9.81m/s. The probability of entrainment P_{s} for a Lagrangian particle for a discrete time-step $dt \in \mathbb{R}$ is as follows (Wang and Shen 2010):

$$P_s = 1 - e^{(-\lambda_{\mathrm{ow}} dt)}. \tag{4.58}$$

If the particle is inserted at this time-step, it enters the water column with intrusion depth:

$$D_i = (1.5 + 0.35(2\phi - 1))H_s, \tag{4.59}$$

where ϕ is a particle specific random variable with a uniform distribution in $[0, 1]$, as described in Delvigne and Sweeney 1988. The maximum depth of intrusion, when $\phi = 1$, can be utilised as z_{crit} to ensure a high resolution grid for entrained sub-surface oil particles.

Oil particle buoyancy is modelled as in Proctor, Flather, and Elliott 1994, whereby oil droplets rise at a constant velocity, determined by the oil droplet size, the water viscosity and the density difference. The droplet distribution is an active research field, with a recent review in Nissanka and Yapa 2018. Though normal, log-normal, Rosin-Rammler and uniform distributions have all been utilised, the review concludes that all distributions are based on an empirical fit, with each distribution the best fit for limited cases. Hence, for simplicity, this work employs a normal distribution fitted to measured data, if available, as in Wang and Shen 2010. If it is not available, then the available distribution for the closest type of oil is utilised until a measurement can be taken. When a particle is entrained into the water column, the particle samples the distribution to determine the droplet size of that particle. The buoyancy velocity is added to w, the vertical component of oil particle velocity. As in Nordam et al. 2019a, particles are considered to have surfaced only if they reach the surface due to buoyancy, without the influence of vertical turbulent diffusion.

4.2.3 Thickness and mechanical spreading

Following the advection, diffusion and entrainment of oil particles, additional particle movement is needed to represent the mechanical spreading of oil if the oil slick thickness is above the terminal spreading thickness of the particular oil. Note that mechanical spreading is applied within each cell, on a cell-by-cell basis and the overall spread of the oil spill is dominated by

the ocean environment processes; advection, turbulent diffusion and stokes drift.

The volume of oil in each thickness map cell is calculated by summing the particles present in the cell, to form $V_{\text{oil}} \coloneqq V_{\text{oil}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$, in units of barrels for the empirical equation. This is then used to calculate area in square meters in Lehr's modified fay-type spreading formula (Lehr et al. 1984), using the lower coefficient for a low wind case (as wind drift is accounted for elsewhere) and the average age of the oil in that cell $t_{\text{oil}} \coloneqq t_{\text{oil}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \in \mathbb{R}_+$ in minutes from the spill start. The empirical slick area $A_{\text{oil}} \coloneqq A_{\text{oil}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ is found by computing

$$A_{\text{oil}} = 10^3 \left(2.27 \frac{\rho_{\text{water}} - \rho_{\text{oil}}}{\rho_{\text{oil}}}^{\frac{2}{3}} V_{\text{oil}}^{\frac{2}{3}} t_{\text{oil}}^{-\frac{1}{2}} + 0.03 \frac{\rho_{\text{water}} - \rho_{\text{oil}}}{\rho_{\text{oil}}}^{\frac{1}{3}} V_{\text{oil}}^{\frac{1}{3}} \left\| \vec{U}_{\text{w}_{\text{knots}}} \right\|_2^{\frac{4}{3}} t_{\text{oil}} \right), \tag{4.60}$$

where $\vec{U}_{\text{w}_{\text{knots}}} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^2$ is the wind velocity converted to knots and where the oil age in hours is $t_{\text{oil}} \in \mathbb{R}_+$. Slick thickness in meters $\Gamma \coloneqq \Gamma(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ in the grid cell of area A_{oil} is then calculated as

$$\Gamma = \frac{V_{\text{oil}_{\text{m}^3}}}{A_{\text{oil}}}, \tag{4.61}$$

where $V_{\text{oil}_{\text{m}^3}} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}_+$ is the volume of oil in the cell converted to cubic meters. Depending on the oil type, if this thickness exceeds that of the equilibrium, or terminal oil thickness then mechanical spreading is applied using Lardners Lagrangian method in the local cell (Lardner and Gunay 2000):

$$\partial Q = 1.13 \left(\frac{\rho_{\text{water}} - \rho_{\text{oil}}}{\rho_{\text{water}}} \right)^{\frac{1}{3}} V_{\text{oil}_{\text{m}^3}}^{\frac{1}{3}} \frac{1}{4} t_{\text{oil}_{\text{sec}}}^{-\frac{3}{4}} dt, \tag{4.62a}$$

$$\partial R = \partial Q + 0.0034 \left\| U_{\text{w}} \right\|_2^{\frac{4}{3}} \frac{3}{4} t_{\text{oil}_{\text{sec}}}^{-\frac{1}{4}} dt, \tag{4.62b}$$

$$x_{\text{new}} = x_0 + \partial Q \cos(\theta_{\text{wind}}) + \partial R \sin(\theta_{\text{wind}}), \tag{4.62c}$$

$$y_{\text{new}} = y_0 + \partial Q \sin(\theta_{\text{wind}}) + \partial R \cos(\theta_{\text{wind}}). \tag{4.62d}$$

For this empirical formula, $t_{\text{oil}_{\text{sec}}} \in \mathbb{R}_+$ is the oil age in seconds and $\theta_{\text{wind}} \coloneqq \theta_{\text{wind}}(x, y, t) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$ is the wind angle, or bearing from north of $\vec{U}_{\text{w}}$. The distances $\partial Q \in \mathbb{R}$ and $\partial R \in \mathbb{R}$ represent the mechanical spreading and the augmented mechanical spreading from wind effects respectively. Equations within Sections 4.2.1, 4.2.2 and 4.2.3 have described the movement of oil particles in the surface and subsurface ocean, but have not accounted for any changing in oil properties through weathering or particle deposition on obstacles and shorelines.

Weathering

A case could be made that for Crude, Heavy or other persistent oils, significant weathering occurs on longer time-scales (a day) than the measuring interval (an hour) and hence can be well accounted for with data-fitting to measurements. The growth in oil droplet size as the oil weathers or ages would be accounted for with droplet distribution fitting. However, with a late sensor arrival, or in some environments (particularly hot, stormy or in the presence of oil dispersant) and for lighter oils, this would be erroneous; with oil properties such as viscosity changing significantly in just a few hours. Hence SCEM can interface with the ADIOS weathering model (Lehr et al. 2002) in GNOME (NOAA 2012), by transferring Lagrangian particles between the two models. The ADIOS weathering model is used to account for emulsification, evaporation and dispersion.

4.2.4 Oil deposition

The model assumes zero particle movement once it enters a non-water cell. If the beach cell is considered saturated, the particle cannot enter (Chao, Shankar, and Wang 2003) and remains afloat. This offers simple shore deposition, though particles cannot re-float once deposited.

4.2.5 Number of oil particles

The presence of random processes modelling oil turbulent diffusion and entrainment cause the spreading of oil particles to become a stochastic process in the simulation. Therefore the number of particles required in the simulation is determined both by the need for accurate reconstruction of a spill shape, and by the need to adequately sample the combined probability distribution function to resolve the process, whichever requires more particles. The stochastic

elements in particle movement are a 2D random walk, a 1D random walk and a dichotomous binomial distribution with a uniform distribution. These are horizontal turbulent diffusion, vertical turbulent diffusion and binary entrainment at a uniformly random depth.

First consider the horizontal turbulent diffusion random walk: Although the distance from origin is not accurately represented by a normal distribution as samples cannot take values less than zero, the distribution of particles along an individual axis can be assumed Normal. The horizontal diffusion normal distribution has the parameters

$$\sigma_{\text{horz}} = (\sqrt{2} - 1)\sqrt{12 D_h dt} \tag{4.63a}$$

and

$$\mu_{\text{horz}} = 0, \tag{4.63b}$$

forming the distribution $N(\mu_{\text{horz}}, \sigma^2_{\text{horz}})$. Define the confidence interval $\alpha_{\text{horz}} \in \mathbb{R}$ and expected random walk movement $E_{\text{horz}} \in \mathbb{R}$ by

$$\alpha_{\text{horz}} = 0.05, \tag{4.64a}$$

and

$$E_{\text{horz}} = \frac{1}{2}\sqrt{12 D_h dt}, \tag{4.64b}$$

then the number of samples needed to approximate the random walk process with a 95% confidence level is (NIST/SEMATECH 2012):

$$n_{\text{horz}} \geq \left(\frac{1.96}{\alpha_{\text{horz}} E_{\text{horz}}} \right)^2 \sigma^2_{\text{horz}}. \tag{4.65}$$

Under typical simulation conditions in Beaufort scale 5 sea states, n_{horz} in (4.65) has a value of approximately 1000, which exceeds the samples needed to approximate the vertical turbulent diffusion, uniform entrainment depth and the number of samples required to apply the central limit theorem to the dichotomous binomial distribution of entrainment. Given the complex interaction between stochastic processes that would greatly increase the variance of the combined probability function, the possibly complex shape of an oil spill, the negligible effect on computational time of increased numbers of oil particles and the implicit desire to improve the

simulation accuracy and confidence limit, it is sufficient that 3000 particles be used. This also exceeds the sum of sample sizes needed for each random process in typical conditions.

4.2.6 The probability of oil presence

This section formulates the probability of oil presence, for use in sensor placement determination. In a simulation realization identified by s_n with $n \in \mathbb{N}$, the presence of oil at a time-step t_k in the cell at position $(x_i, y_j, z_w) \in \Omega$ is described by the binary random variable $O_p(x_i, y_j, z_w, t_k, s_n)$, which takes the value 0 when the oil volume in the cell at (x_i, y_j, z_w) is less than an arbitrary threshold value $\zeta_p \in \mathbb{R}_+$ (no oil present) and the value 1 when the oil volume in the cell is greater than ζ_p (oil is present), at time-step t_k. The binary random variable is described by

$$O_p(x_i, y_j, z_w, t_k, s_n) = \begin{cases} 0 \text{ when } \tilde{V}_{\text{oil}}(x_i, y_j, z_w, t_k, s_n) \leq \zeta_p, \\ 1 \text{ when } \tilde{V}_{\text{oil}}(x_i, y_j, z_w, t_k, s_n) > \zeta_p, \end{cases} \tag{4.66}$$

where the function $\tilde{V}_{\text{oil}}(x_i, y_j, z_w, t_k, s_n) : \Omega \times \mathbb{R}_+ \times \mathbb{N}_+ \to \mathbb{R}_+$ returns the volume of oil present in the discrete cell (x_i, y_j, z_w) at time t_k for realization s_n. Consequently, the evolution of oil presence across the spatial domain is described by the stochastic process $\{O_p(\Omega, t_k, s_n)\}_{t_k \in \mathbb{R}_+}$, the set of binary random variables describing oil presence in the spatial domain Ω for each time-step t_k, for realization s_n. The presence of oil in a set of space and time $A \subseteq \Omega \times \mathbb{R}_+$ that may span multiple time-steps, on a discrete mesh, is characterised by the binary random variable

$$\tilde{O}_p(A, s_n) = \begin{cases} 0 \text{ when } \sum\limits_{x_i, y_j, z_w, t_k \in A} O_p(x_i, y_j, z_w, t_k, s_n) = 0, \\ 1 \text{ when } \sum\limits_{x_i, y_j, z_w, t_k \in A} O_p(x_i, y_j, z_w, t_k, s_n) \neq 0. \end{cases} \tag{4.67}$$

Hence, $\tilde{O}_p(A, s_n)$ only takes value 0 if the volume oil in every cell is less than or equal to ζ_p for the entire spatio-temporal set A, or takes value 1 if the oil volume in any cell exceeds ζ_p at any time, in the realization s_n.

To inform sensor placement it is useful to describe the probability of oil presence in A, by utilising multiple realizations each of which is assumed to be an independent stochastic process. Multiple realizations are needed to examine model sensitivity to uncertain parameters, such

as drift coefficients. The probability of oil volume exceeding ζ_p using $s_T \in \mathbb{N}$ realizations, $\mathbb{P}[\hat{O}_p(A) = 1, s_T]$ is defined by

$$\mathbb{P}[\hat{O}_p(A) = 1, s_T] = \frac{\sum_{s_n=1}^{s_n=s_T} \tilde{O}_p(A, s_n)}{s_T}. \tag{4.68}$$

To determine the number of realizations needed to adequately sample the random processes, the probability of oil presence sample variance after s_T realizations is calculated (Montgomery and Runger 1994) by

$$\mathrm{Var}(\mathbb{P}[\hat{O}_p(A) = 1]) = \frac{1}{s_T - 1} \sum_{s_n=1}^{s_n=s_T} \left(\mathbb{P}[\hat{O}_p(A) = 1, s_n] - \bar{\mathbb{P}}[\hat{O}_p(A) = 1, s_n] \right)^2, \tag{4.69a}$$

with a maximum value across Ω of

$$\mathrm{Var}_{\max}(\mathbb{P}[\hat{O}_p(A) = 1]) = \max_{A \in \Omega} \left(\mathrm{Var}(\mathbb{P}[\hat{O}_p(A) = 1] \right), \tag{4.69b}$$

where $\bar{\mathbb{P}}[\hat{O}_p(A) = 1, s_T] = \frac{1}{s_T} \sum_{s_n=1}^{s_n=s_T} \mathbb{P}[\hat{O}_p(A) = 1, s_n]$ is the mean probability of oil presence for s_T realizations. For the parameters of Table 4.1 and an oil threshold value of $\zeta_p = 0$, the maximum value of the variance (4.69b) with realization number decreases rapidly, then settles after $s_n \approxeq 200$ as in Figure 4.10. The variance distribution of (4.69a) displayed peaks at the trail and leading edges of the spill, as expected due to the changing in presence of oil across realizations compared to the overlap of spills at the spill centre. The variance in oil presence probability is utilised instead of the variance in oil presence, as a confidence interval in oil probability describes a range of chance in oil presence and is more useful than a confidence interval describing the range of realizations with oil present. A 3 sigma confidence interval from (4.69a) would be, for example, a statement that there is a probability of 99.7% that the probability of oil in a given area is between determined lower and upper bounds. In practice, this would be used to identify oil-free and oil-likely areas with some degree of confidence.

In a simulation realisation denoted by s_n with $n \in \mathbb{N}$, for a selected oil particle p_i, with $i \in \mathbb{N}$, a particle has the 2D position denoted $\vec{O}(p_i, t, s_n)$ where $\vec{O} : \mathbb{N} \times \mathbb{R}_+ \times \mathbb{N} \to \mathbb{R}^2$ and is modelled as a vector valued random process.

4.2.7 The probability of oil drift location at a specific time

Another useful event to model, and for sensor placement determination, is the surface location of a selected oil volume at a given time-step. Define the vector valued random variable $O(p_i, t_k, s_n)$ where $\vec{O} : \mathbb{N} \times \mathbb{R}_+ \times \mathbb{N} \to \mathbb{R}^2$, to take the vector position of the oil particle index $p_i \in \mathbb{N}$ at the time-step t_k, with $x_p \in \mathbb{R}$ and $y_p \in \mathbb{R}$ the horizontal and vertical locations respectively, for a given realization s_n. The probability of an oil particle p_i to be within the discrete cell (x_i, y_j) at t_k, $\mathbb{P}[O(p_i, t_k, s_n) \in (x_i, y_j)]$, is defined by

$$\mathbb{P}[O(p_i, t_k, s_n) \in (x_i, y_j)] = \frac{\sum\limits_{p_i \in p_p(x_i, y_j, t_k, s_n)} V_{\text{particle}}(p_i, s_n)}{\sum\limits_{p_i \in p_t(t_k, s_n)} V_{\text{particle}}(p_i, s_n)}, \tag{4.70}$$

where $p_p(x_i, y_j, t_k, s_n) : \partial\Omega \times \mathbb{R}_+ \to \mathbb{N}^{m_p}$ is a vector of particle indices present in the discrete spatio-temporal location and $p_t(t_k, s_n) \to \mathbb{N}^{m_t}$ is a vector of all particle indices at time t_k, with m_p and m_t being the number of oil particles present and the total number of oil particles respectively. The oil volume function $V_{\text{particle}}(p_i, s_n) : \mathbb{N}^{m_t} \to \mathbb{R}_+$ maps oil particle indices p_i to the oil volume they represent in the model. Evaluation of (4.70) for every cell in $\partial\Omega$ forms the probability mass function displayed in Figures 4.8 and 4.9. The probability of oil presence in cell (x_i, y_j) is obtained by averaging over the realizations of the stochastic process. The resulting probability is given by

$$\mathbb{P}[\hat{O}(p_i, t_k) \in (x_i, y_j)] = \frac{1}{s_T} \sum_{s_n=1}^{s_n=s_T} \mathbb{P}[O(p_i, t_k, s_n) \in (x_i, y_j)], \tag{4.71}$$

where $\mathbb{P}[O(p_i, t_k, s_n) \in (x_i, y_j)]$ is the evaluation of (4.70) for that realization index. This probability, $\mathbb{P}[\hat{O}(p_i, t_k) \in (x_i, y_j)]$, provides a further measure for route planning by indicating likely areas of high oil volume, while the probability of oil presence $\mathbb{P}[O_p(x_i, y_j, z_w, t_k) = 1]$ defines likely areas of any oil exceeding a threshold ζ_p.

4.2.8 The use of probability of oil drift and presence in sensor path optimisation

The uncertainty minimisation weighting across the spatio-temporal domain is a function of oil drift, or oil presence probability, amongst others, detailed in Chapter 5. Given the above sections on probability, it should be noted that the probability of oil presence $\mathbb{P}[\hat{O}_p(A) = 1, s_T]$ in a cell (x_i, y_j) is likely highly inaccurate when the number of simulations s_T sampling from a distribution of uncertain parameters is very small. Note in Figure 4.10 the sharp fall off of oil presence variance and hence the choice of $s_T = 20$ as the switching point from minimisation weighting using oil drift probability to oil presence probability. The oil drift probability is initially used to ensure spatial variation in the selected probability measure, and oil drift probability is the only suitable measure when $s_T = 1$.

The min-max normalised over $\partial\Omega$ probability of oil presence or drift for a cell (x_i, y_j), at a given time, is defined by $P_{\hat{\bar{O}}_p}(x_i, y_j)$ where $P_{\hat{\bar{O}}_p} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \to [0, 1]$. If $s_T > 20$ this is the min-max normalisation of oil presence probability $\mathbb{P}[\hat{O}_p(A) = 1, s_T]$ where A is area of the cell (x_i, y_j). However, when $s_T \leq 20$, $P_{\hat{\bar{O}}_p}(x_i, y_j)$ is defined as the min-max normalised over $\partial\Omega$ probability of oil particle drift to cell (x_i, y_j), at a given time, instead. This is the min-max normalised value of $\mathbb{P}[\hat{O}(p_i, t_k) \in (x_i, y_j)]$.

4.2.9 The mean location of the spill centre

A further vector valued random variable $O_m(t_k, s_n) = [x_m, y_m] \in \partial\Omega$ takes the value of the position of highest oil volume for realization s_n, where $x_m \in \mathbb{R}$ and $y_m \in \mathbb{R}$ are the horizontal and vertical locations of the highest volume position respectively. The value taken by $O_m(t_k, s_n)$ is one definition of the spill centre. Define the mean spill centre position across realizations by

$$\bar{O}_m(t_k) = \frac{1}{s_T} \sum_{s_n=1}^{s_n=s_T} O_m(t_k, s_n). \tag{4.72}$$

The mean spill centre position for 500 realizations of the Grande America simulation in Section 4.4 is displayed in Figure 4.11.

4.3 Backwards SCEM model

In order to offer backtracking or hind-casting capabilities for source location, the model must be capable of simulating a detected oil spill advecting backwards in time, and converging if there is an assumption of a point source. The location of the spill at the estimated spill start time is an estimated source location.

There is a lack of determinism in reverse time without modification of the forward model (Galt and Payton 2005; Ciappa Achille; Costabile 2014), in all non-linear or stochastic processes forming the oil model, and this is why prior work assesses possible starting points by running a forward simulation and compares the result, see Breivik et al. 2012. However, a forward ensemble approach requires multiple simulations and may be ill-suited for time constrained simulations. The short-time frame of simulations here may allow a single backward simulation using modified algorithms to produce an accurate source estimation, or at least, a refined set of source locations for an ensemble of forward simulations.

4.3.1 Ocean model

Due to the ill-posed nature of backwards simulation of the Navier-Stokes equations, the ocean model runs forward from the oldest relevant data at $t = 0$ to the newest at $t = t_f$ and stores the data required by the oil model. The sub-surface velocities are extrapolated from the surface velocities, wind speeds and the averaged wind speed for the Ekman current, hence only these values are stored. The wave spectra are calculated by SWEM using the surface current and wind speeds at each time-step and are not past-time-dependent, hence they do not need to be stored.

4.3.2 Oil model

The oil model steps backwards from t_f loading the velocities from the ocean model, calculating the sub-surface velocities and the wave spectrum in SWEM with its Stokes drift velocity, then applying the oil processes with a negative time-step. The negative time-steps continue until the estimated $t = 0$, or until the oil reaches an estimated starting thickness or size. The oil processes for entrainment, diffusion, mechanical spreading are also ill-posed in reverse: An example would be that, forwards in time, an oil patch is entrained into the water column in a

patch of rough sea. The oil enters sheared flow, moves underwater and then rises to the surface to form a new patch in calmer waters. In a backwards time simulation, starting with the oil patch in calmer waters, nothing is entrained into the water column and so the patch does not advect backwards through the sheared flow and is at the incorrect position at the starting time.

Sub-surface oil particles

Oil particles that begin entrained into the water column undergo negative buoyancy until they sink to the intrusion depth of their current cell, where upon they rise to the surface and become part of the main spill. Otherwise, oil particle entrainment is disabled, as wave induced entrainment is ill-posed in a reverse simulation. Vertical turbulent diffusion is left as a random process, and buoyancy remains positive for all non-entrained particles; this ensures the spills vertical dispersion near the surface is consistent, albeit without entrainment.

Turbulent diffusion

The random walk process that describes the forward horizontal turbulent diffusion would still be a random process with the time-step inverted. The expected motion of the random walk process needs to be examined to infer a reverse motion, where particles will be required to step towards a position, likely the centre of the oil spill, to simulate reverse diffusion. The forwards random walk has both a uniform step size distribution, between 0 and $\sqrt{12D_h dt}$ and a uniform direction distribution between 0 and 2π, see Section 4.2.1. For a uniform, constant step size, it is known (Weisstein 2017) that for particle counts tending to infinity:

$$S_{\text{rms}} = l\sqrt{N} \tag{4.73}$$

where $S_{\text{rms}} \in \mathbb{R}_+$ is the root-mean-squared distance travelled, $l \in \mathbb{R}_+$ is the step size and $N \in \mathbb{N}$ is the number of steps. For a uniform step size distribution and time-step δt, the expected value $E_{\text{step}} \in \mathbb{R}_+$ can be substituted for the step size, calculated by:

$$E_{\text{step}} = \frac{1}{2}\sqrt{12D_h\delta t} \tag{4.74}$$

By substituting the number of steps for $N = t_{\text{oil}_{\text{sec}}}/\delta t$, where $t_{\text{oil}_{\text{sec}}}$ is the age of the oil spill in seconds, the dependency on the time-step can be removed, as the full equation simplifies:

$$S_{\text{rms}} = \frac{1}{2}\sqrt{12D_h\delta t}\sqrt{\frac{t_{\text{oil}_{\text{sec}}}}{\delta t}}, \tag{4.75a}$$

$$S_{\text{rms}} = \frac{1}{2}\sqrt{12D_h t_{\text{oil}_{\text{sec}}}}. \tag{4.75b}$$

For an estimated boundary radius, the r.m.s distance can be converted to a peak to peak value, using the maximum step size of $2E_{\text{step}}$, by

$$S_{\text{pk}} = 2\sqrt{2}\sqrt{12D_h t_{\text{oil}_{\text{sec}}}}. \tag{4.76}$$

However, this has an accuracy limitation, since it operates under the assumption that the oil has been subject to this D_h for all of $t_{\text{oil}_{\text{sec}}}$ and D_h is not uniform across the spatio-temporal domain. The derivative of Equation (4.76) gives a shrinking velocity $U_{\text{shr}} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^2$, the speed at which particles on the boundary of the oil spill should move towards the centre. This is scaled by the distance a particle is from the spil centre, $S_{\text{particle}} \in \mathbb{R}_+$, as a proportion of the S_{pk} radius. The scaling factor dependence on the S_{pk} represents an estimate of the spill radius resulting from turbulent diffusion alone. To combat the over-estimation of shrinkage on inner particles, shrinkage is only applied to particles outside the estimated radius when $S_{\text{particle}} > S_{\text{pk}}$. The shrinkage velocity U_{shr} is described by,

$$U_{\text{shr}} = \frac{2\sqrt{6}D_h}{D_h t_{\text{oil}_{\text{sec}}}} \frac{S_{\text{particle}}}{\frac{1}{2}S_{\text{pk}}}. \tag{4.77}$$

Hence, the backwards turbulent diffusivity velocity can be determined by

$$\begin{bmatrix} u' \\ v' \end{bmatrix} = U_{\text{shr}} \begin{bmatrix} u* \\ v* \end{bmatrix}, \tag{4.78}$$

where $u*$ and $v*$ are the horizontal and vertical components of the unit vector from the particle position to the spill centre.

Mechanical spreading

Mechanical spreading is again, ill-posed in reverse and is dependent upon environmental conditions (wind speed etc) as well as the oil thickness in the forwards simulation. Consideration has been given to implementing an empirical shrinkage, similar to turbulent dispersion. An estimated ellipse from mechanical spreading (Chao, Shankar, and Wang 2003) is described by

$$L_{\min} = 53.76 \left(\frac{\rho_w - \rho_o}{\rho_o} \right)^{1/3} V_{\text{oil}}^{1/3} t_{\text{oil}}^{1/4}, \tag{4.79a}$$

$$L_{\max} = L_{\min} + 0.95 U_{\text{wind}}^{4/3} t_{\text{oil}}^{3/4}, \tag{4.79b}$$

$$A = (\pi/4) L_{\min} L_{\max}. \tag{4.79c}$$

In a similar manner to the turbulent diffusion, the derivative of equations (4.79a) provides a mechanical spread shrinkage velocity, to be scaled by the particles positions along the ellipses long axis $L_{\max}$, towards the spill centre. Again, to combat the over-estimation of shrinkage on inner particles, spill shrinkage is only applied to particles outside the estimated radius. Although the long axis includes effects of wind, using the short axis led to the shrinkage velocity being too high in simulations.

4.3.3 Practical considerations for source locating

After investigation through backtracking of spill particles released from a known point source, the above methods of spill shrinkage are interesting, but ultimately not useful for source locating. The methods force convergence of a spill and give some indication of a spill size when stepping a spill backwards, but are only valid for a roughly elliptical spill shape with a well defined centre and axis lengths. Furthermore, due to many not-modelled processes when running backwards in time, the location at which the spill converges to a point is very seldom the true source location, though it may be close. In practice, it is preferred to advect and diffuse the spill backwards in time, omitting ill-posed processes, with further diffusion based spreading and accept that there will be a large area identified as possible sources at the estimated release

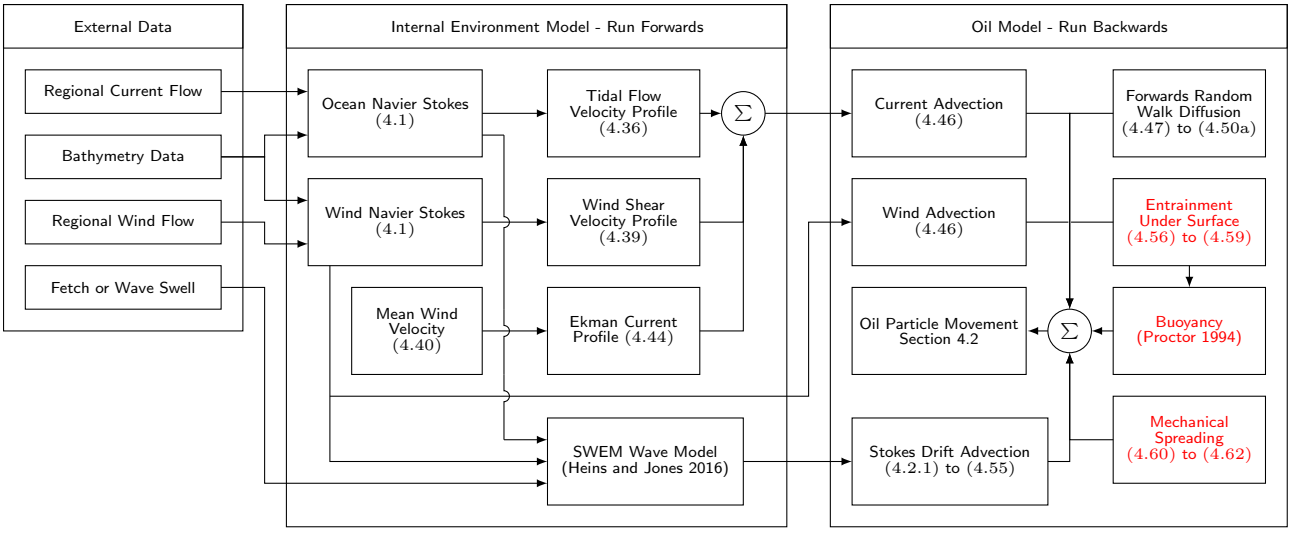


Figure 4.7: A block diagram of the combined fluid and oil model, in the backwards mode. Note that the fluid model runs forward in time, to produce a fluid state trajectory, that the oil model then steps through backwards in time, with the SWEM wave spectra only being calculated when needed. When running backwards the oil model omits the processes of entrainment, buoyancy and mechanical spreading, highlighted in red. Note that the random walk diffusion continues to spread a spill out as it is advected backwards.

time. This backwards implementation of SWEM is displayed in Figure 4.7.

As mentioned in Ciappa Achille; Costabile 2014, the random walk backwards in time describes the uncertainty of the trajectory caused by turbulent diffusion, while the backwards in time integration of advection provides probabilities of possible source locations. Then, similar to prior work (Breivik et al. 2012), run an ensemble of spill releases from the possible source locations, then ascribe a probability to each source location using the prediction-to-measurement correlation of the spatio-temporal distribution of oil volume for each spill.

In the source locating scenario in Chapter 7, the source locating process is described thus: After a suitable amount of sensing time, 12 hours, 20 hours and 24 hours, an analysis model run occurs. This assimilates all the fluid measurements up to that sensing time, to produce an improved analysis state-trajectory for the ocean model. Then, running back-in-time from the latest measurement, spill particles are injected into the ocean only where they have been measured, and backtracked with advection and diffusion through the ocean flow. Though this reverse time advection and diffusion is strictly speaking, ill-posed, the distribution of particles at their supposed leak-times identifies a range of possible source locations and times. An ensemble of forward spills are then simulated from each of these possible source locations and times. As the ensemble particles move forward through time, they are compared with the

sensor measurements of oil presence. If the particles from a source location are present when an oil measurement confirms oil, then the source location receives +1 correlation score. If they are not present when an oil measurement confirms oil, then the source location receives −1 correlation score. The source location with the highest score after all sensor measurements are taken into account, is the identified source location to be used for an analysis run of the oil model, with the analysis run of the ocean model.

4.4 Model simulation and results

The model is intended to guide sensing assets in the aftermath of maritime incidents and hence requires validation, with comparison against real-world data preferable (Spaulding 2017). The Grande America oil spill of March 2019 provides a recent and observed incident to validate against. However due to the vessel's abandonment on the 11th March 2019 due to an on-board fire and the subsequent sinking in water depth of 4600 m between 1500 and 1800 hours on the 12th March 2019, it is unclear exactly when the vessel sank, the oil leak occurred, or how much leaked. This information forms the initial conditions for the spill and can heavily affect simulation results.

For the model simulation it is assumed the fuel tanks became compromised as the hull split and sank and the worst case scenario is modelled: all 2200 tonnes of Heavy Fuel Oil carried by the Grande America is spilt in a short time-frame, from 1400 to 1600 hours on the 12th March 2019, at coordinates -5.7844° East, 46.0689° North. The model utilises Global Forecast System (GFS) wind velocities and Tidetech ocean velocities, with a North-West to South-East wave swell with significant wave height of 3m from National Centers for Environmental Prediction (NCEP) data. User specified parameters are presented in Table 4.1. The Grande America oil spill was observed by the Copernicus Sentinel 1 and 2 satellites on two occasions, on the 19th March 2019 the 5 day old slick is observed at approximately 45.439458° North, -4.283424 ° East and on the 23rd March 2019 the 11 day old slick is observed at 45.0826° North, -4.4559° East. Due to the uncertainty surrounding the initial spill conditions and volume, emphasis is placed on the model accurately predicting the drift of a spill, with little importance placed on predicting the slick thickness or volume. The probability of oil drift location in a spatio-temporal domain is given by equation (4.70), this is evaluated for each grid-cell area at the indicated time to

produce Figures 4.8 and 4.9. Figures 4.8 and 4.9 show accurate prediction of the slick locations, with high probability at 45.2000° North, -4.1850° East on the 19th March 2019, with the true location being 45.1857° North, -4.323424 ° East, ≈14km to the north west. For the 23rd March 2019, with no correction or reinitialisation from the true spill position on the 19th March, the model predicts a slick location at 45.0300° North, -4.2100 ° East, compared to the true position at 45.0826° North, -4.6559° East, 20 km to the west of the predicted position. Errors of 15 km and 20 km for five and eleven day predictions, respectively, not unreasonable given the scale of the spill, the large size of the domain, the lack of model correction or calibration and the model's intended purpose for predictions over much shorter time-scales (hours to a day). The 288 hour prediction took 568 seconds to compute in MATLAB, on a Windows 10, i7-6700k CPU desktop computer, this includes computation time for the wind, wave and ocean hydrodynamic models across 2688 surface nodes, extrapolated to 534912 sub-surface nodes. All parameters were within ranges acceptable to literature and use their values within Table 4.1.

To investigate the sensitivity to the spill parameters and the diffusion, wind and ocean current coefficients of equation (4.46), 500 simulation realizations using simultaneous sampling of the random variable coefficients of Table 4.1 were utilised to get a probability of oil presence map (4.68) across the set of random variables. Figure 4.11 shows that the model is accurate for the Grande America spill within the typical bounds for drift parameters and that the default coefficient values slightly overestimate oil movement up to the 19th March 2019 when comparing the results of Figures 4.8 and 4.11.

As demonstrated by the Grande America spill, data is scarce on spill components, environmental data, contaminant position and thickness following a real incident, with even international scale incidents only becoming well observed and documented several days after the incident.

The model is now compared to the industry standard model GNOME as a benchmark to see if it provides similar predictions when provided with the same inputs. The comparison simulation utilises Global Forecast System (GFS) wind velocities and Tidetech ocean velocities. As large data set acquisition is unlikely to be available in deployment due to data transmission constraints, the Tidetech ocean data results are assumed to be available as surface velocity only. The data set includes Ekman currents and therefore Ekman currents are omitted from the surface dynamics, but a spiral is calculated sub-surface to model oil slick shear separation

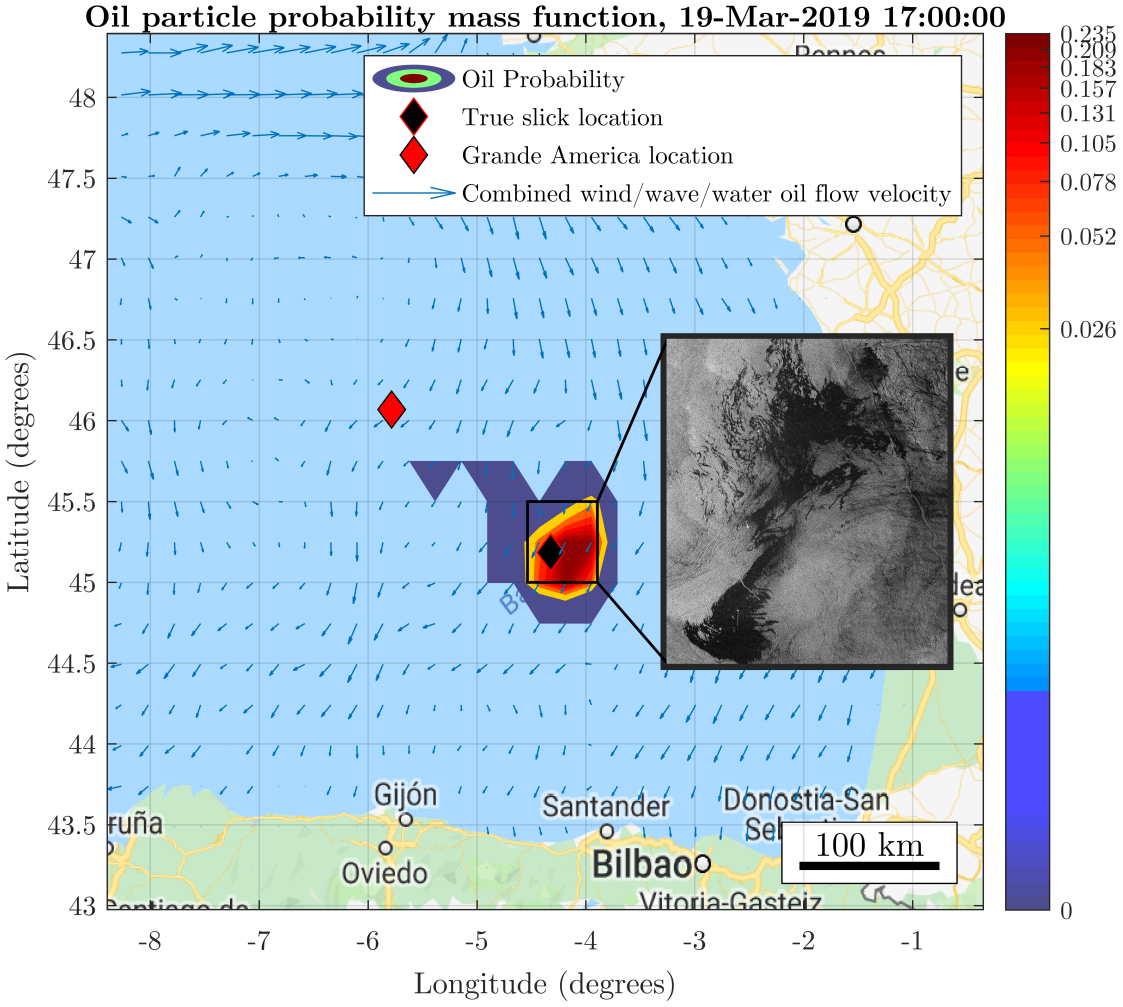


Figure 4.8: The probability mass function of oil particle drift (4.71) from the SCEM simulation for 17:00 on 19th March 2019, 5 days after the spill released, using a log scale and with the real position marked. Note the similarity in location to the real slick location on the 19th March. Map data ©2019 Google, Inst. Geogr. Nacional. Contains modified Copernicus Sentinel data (2019), processed by ESA, CC BY-SA 3.0 IGO.

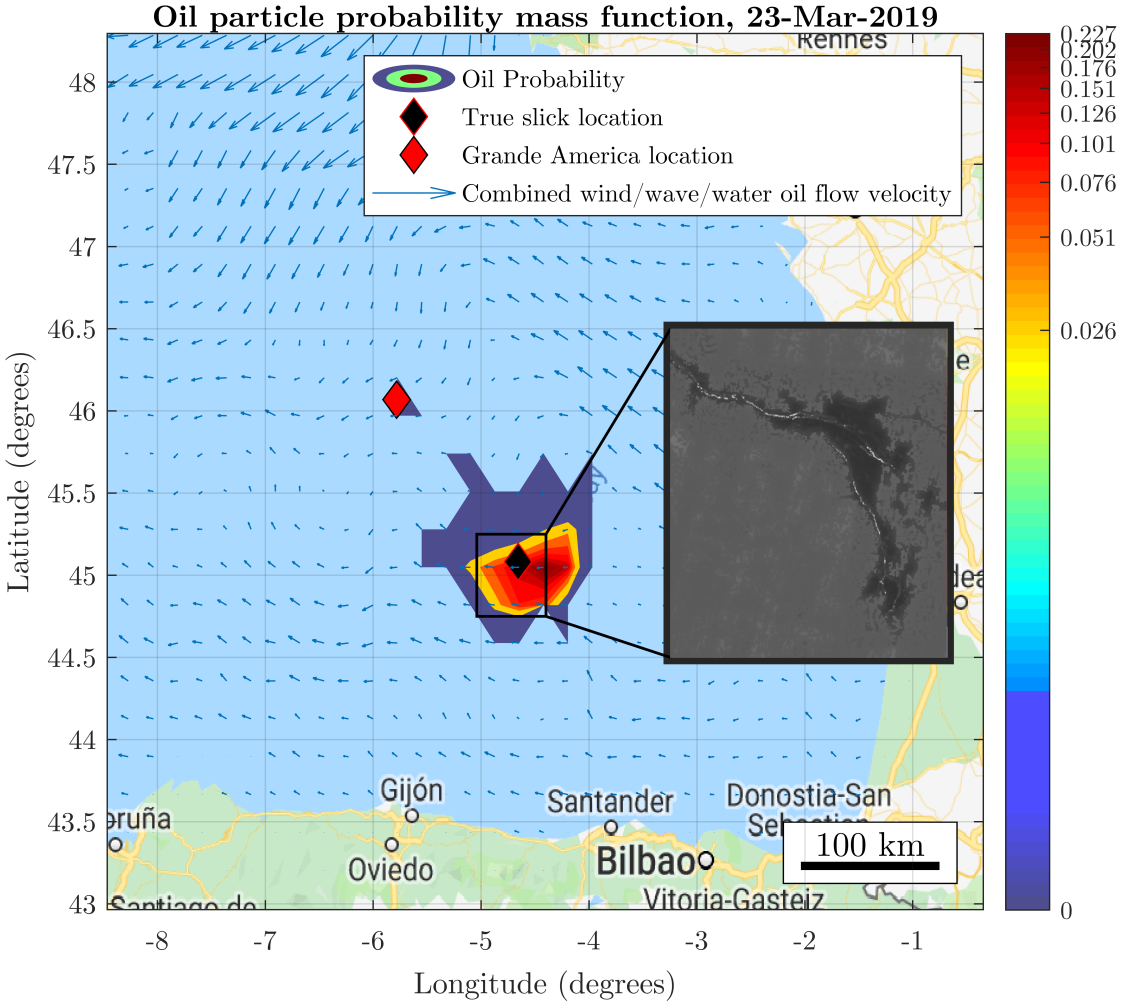


Figure 4.9: The probability mass function of oil particle drift (4.70) from the SCEM simulation for the 23rd March 2019, 11 days after the spill released, using a log scale and with the real position marked. Note the similarity in location to the real slick location on the 23rd March. Map data ©2019 Google, Inst. Geogr. Nacional. Contains modified Copernicus Sentinel data (2019), processed by ESA, CC BY-SA 3.0 IGO.

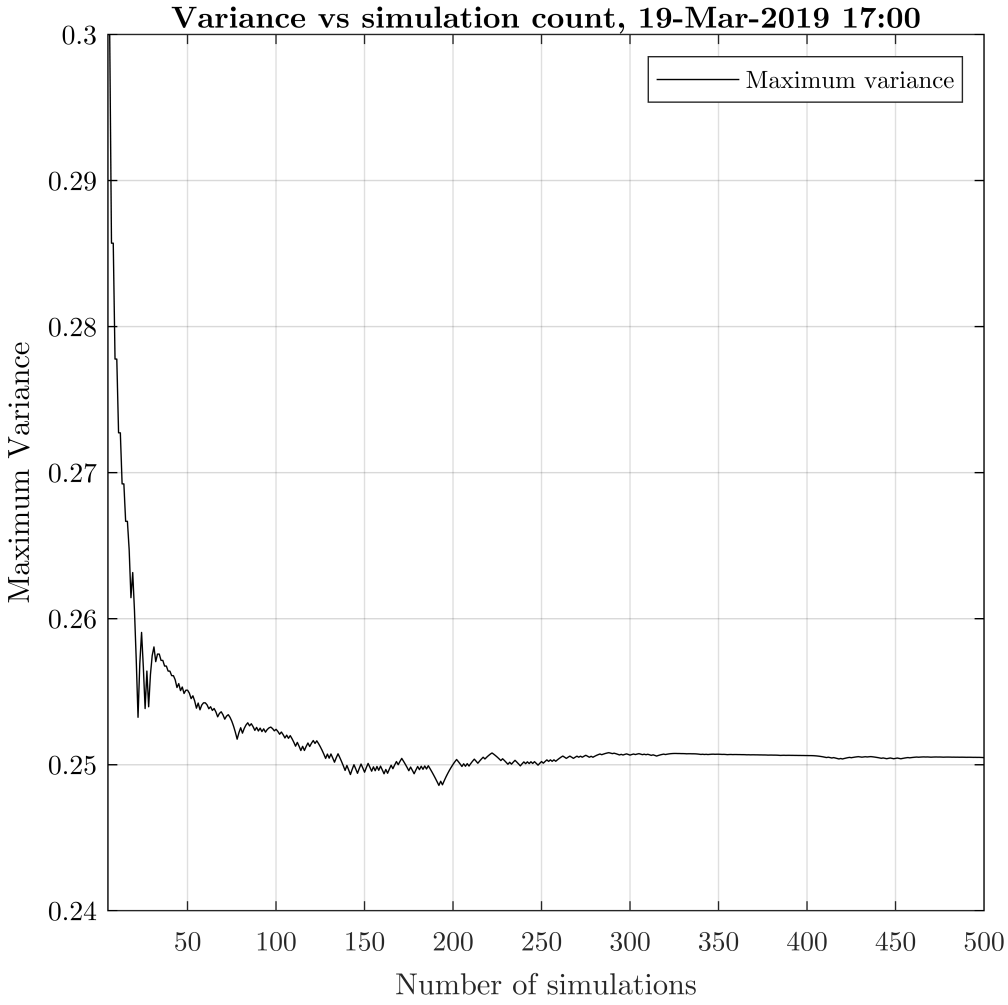


Figure 4.10: The plot showing the decay of the maximum variance of oil presence (4.69b) across 500 SCEM simulations for 17:00 on the 19th March 2019, 5 days after the spill released. Note the rapid decay and convergence, settling around 200 simulations.

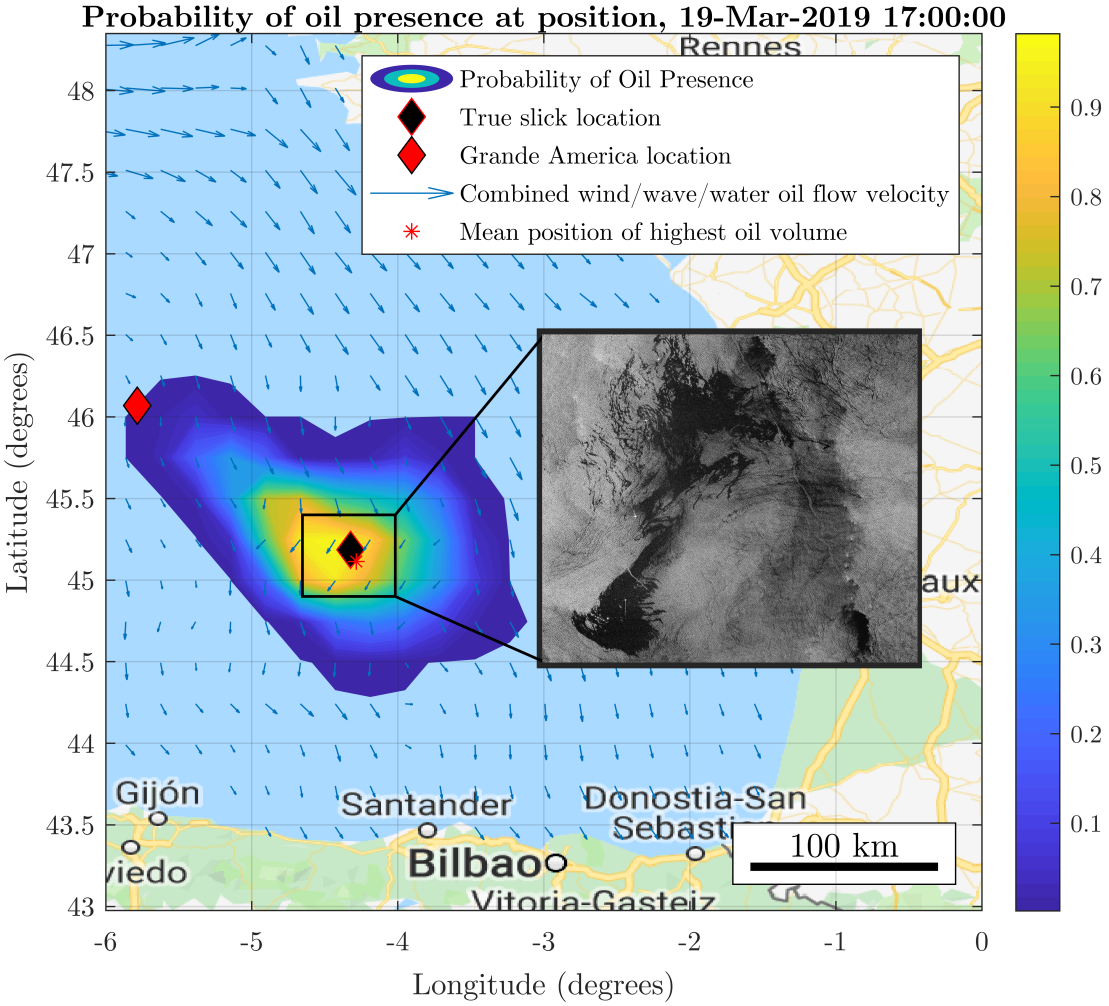


Figure 4.11: The combined probability map of oil presence (4.68) for 17:00 on the 19th March 2019, 5 days after the spill released, with the real position marked. Each SCEM simulation was a sampling of spill parameters in Table 4.1. Note the similarity in location of the highest probability and mean spill position to the real slick location on the 19th March. Map data ©2019 Google, Inst. Geogr. Nacional. Contains modified Copernicus Sentinel data (2019), processed by ESA, CC BY-SA 3.0 IGO.

between surface and sub-surface particles. Both models utilise the same number of particles, representing the same volume of oil each and released at the same leak rate from the same location. Therefore, particle positions can be utilised for comparative purposes.

The model comparison is a 3-day simulation of a 100 barrel spill released 1 mile south of Lamma Island, Hong Kong, at 0330 hours on the 8th January 2019 carried out as a contingency for the Aulac Fortune oil tanker explosion. The oil models are both forced by Tidetech ocean current data and GFS wind data and their similarity displayed in Figure 4.12 indicates SCEM is at least a comparable model that includes the resolving of external data in the environment model (Section 4.1).

4.5 Uncertainty and sensor modelling

This section first discusses the key information required for monitoring and litigating a maritime incident, then formulates a description of uncertainty, to be later minimised in a cost function in Chapter 5. Recall from Chapter 1 the overview of a maritime incident, the typical response and the intended use of information gathered to support guide further monitoring, clean-up operations and provide evidence for litigation purposes. Here, there is an assumption that sensors are only capable of measuring environment and oil properties at the ocean surface, limited to the wind and ocean flow velocity, the peak wave amplitude and period, and the oil thickness. To briefly summarise the review in Chapter 2, two principal approaches are favoured: those of placing sensors in locations that maximise the expected improvement in estimation or expected information utility (Morlier et al. 2018; Hutchinson, Liu, and Chen 2019), or minimise the trace of the estimation error covariance (Zammit Mangion, Anderson, and Kadirkamanathan 2011; Yan et al. 2019). Here, the uncertainty minimisation approach is taken: A design choice is made that gaining knowledge of uncertain variables is most important, to the extent it should almost be the sole focus through uncertainty minimisation. In a maritime incident where there are large sources of uncertainty inherent in both the scenario and the mathematical descriptions, mapping the spill then establishing a broad description of the ocean and oil dynamics for extrapolation is far more important than model refinement; at least until the situation is well estimated. However, the spatio-temporal weighting of uncertainty minimisation by measures of information utility is used here to balance sensor focus between

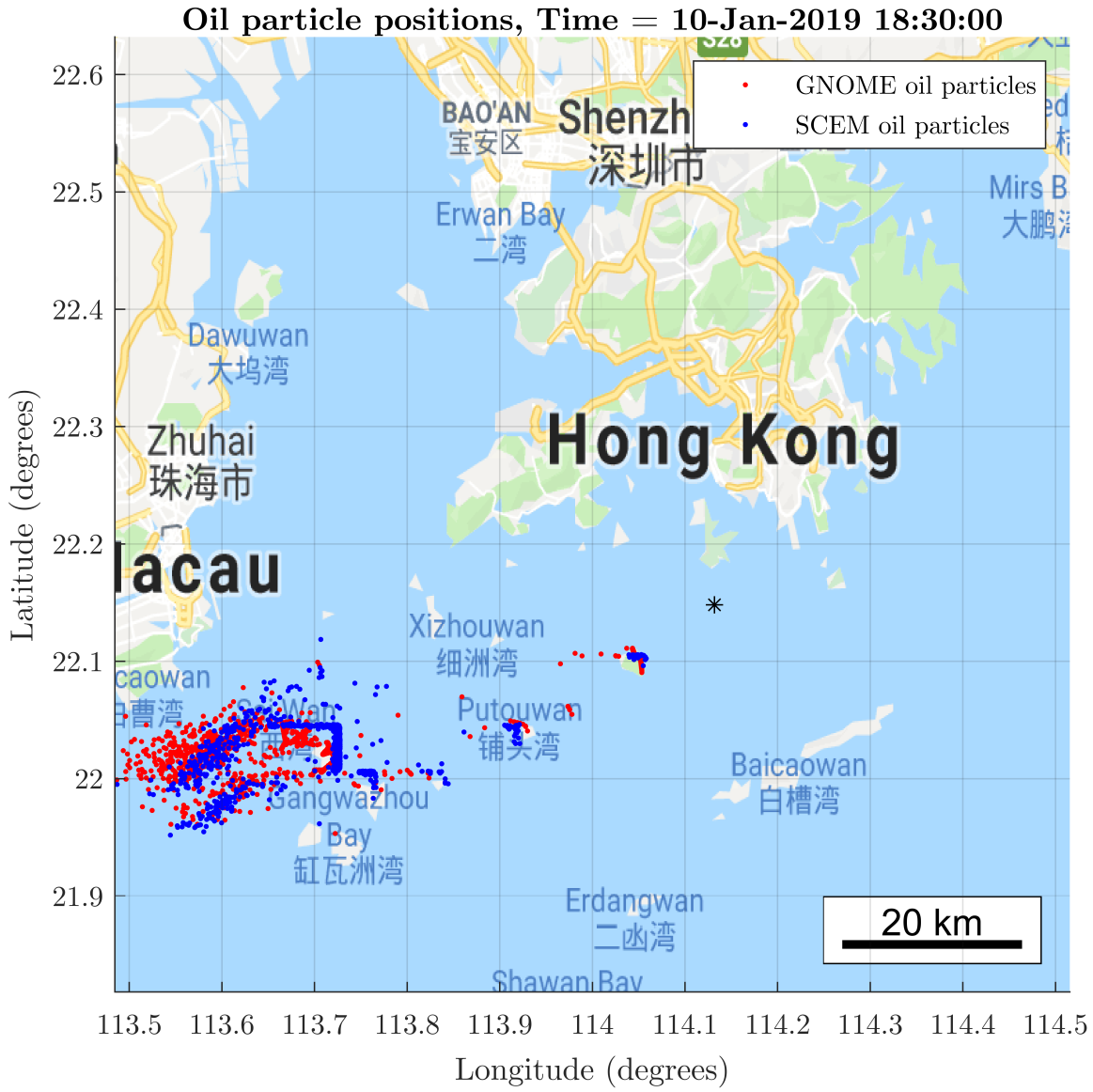


Figure 4.12: The model and GNOME simulation results for a 3-day simulation of a 100 barrel spill released 1 mile south of Lamma Island, Hong Kong, at 0330 hours on the 8th January 2019. Both models has been forced by GFS wind velocities and Tidetech ocean velocities. Note the presence of oil on all the same islands and positions of the leading edge of the spill. Map data ©2019 Google, Inst. Geogr. Nacional.

confirmation (also termed exploitation) and exploration.

In this scenario, the first key piece of information is the chance of oil presence in an area, throughout the temporal domain. This is determining where the oil was in the past, including the source location, where the oil is now and where the oil is going to be in the future. This is described by the probability of oil particle presence in SCEM, the combined ocean and oil model, within a discrete area (4.68). The second key piece of information is the distribution of oil within the spatial domain, throughout the temporal domain, which can be used to describe the oil concentration and thickness. This is described by the probability of oil particle drift location in SCEM within a discrete area (4.71). It would now be useful to quantify the uncertainty of these two key probabilities and place sensors to minimise the defined uncertainty. There is prior work in Goncalves et al. 2016 to formulate descriptions of uncertainty for an oil model using a polynomial chaos approach. However, this approach relies upon constructing a surrogate model and a large ensemble (1024 model realisations) with which to form a robust estimate of the models statistical distribution. As discussed in Chapter 2, a large ensemble is not feasible given the time constraints on the computation time and hence the use of (4.69a) is impractical in a control setting. Here, it is desired to produce a definition of uncertainty specific to surface measurements of oil and environment parameters that does not require multiple simulation runs, and this uncertainty will be a key component in the sensor pathing algorithm described in Chapter 5.

A complete system description convenient for state estimation with a Kalman filter or Gaussian-Process-Regression is not possible due to the non-linearities, stochastic processes and high number of states present in the full combined ocean and oil model description. Hence, the use of the estimation error covariance matrix as a measure of uncertainty (as in Zammit Mangion, Anderson, and Kadirkamanathan 2011) is not feasible here. The method employed instead is to construct a secondary system of PDEs that approximates the uncertainty of the oil model surfaces, described in this chapter. Then construct a reduced order model of the ocean model suitable for state estimation with a Kalman filter, described in Chapter 6, with the estimation uncertainty as a function of the standard estimated error covariance matrix and reduced order modelling error. It is apparent that the uncertainty in estimating ocean and wind flow dynamics also introduces uncertainty in the oil model, so these two descriptions of uncertainty are

combined to form the definition in this chapter.

Note that the important outputs, (4.68) and (4.71), along with oil thickness, are dependent upon the position of particles in SCEM and hence the principal uncertainty in the key outputs of SCEM can be described through the uncertainty in particle position in SCEM. Before continuing, it is important to establish the meaning of uncertainty, as there are two common measures. They are the variance and discrete entropy of a variable. Note that entropy is defined for continuous distributions but its definition has a number of short-comings (Marsh 2013). The variance captures the spread of outcomes quantitatively as a deviation around the mean value, while entropy is something that maximises when each outcome occurs with the same probability (large uncertainty) and minimises when there is only a single outcome (no uncertainty). Choosing which measure to utilise is not entirely simple: In the case of a multi-modal distribution where the probability density function contains multiple peaks, the variance is a poor measure as it fails to present this important structural information. In this case, entropy would increase to describe the multi-modal nature. However, entropy is independent to the range of outcomes as it is solely dependent upon the probability of each outcome. In this case, the range of outcomes is important: consider a random variable, e.g. representing particle position on an axis, that samples two discrete distributions. The first distribution has equal probability of outcomes 0 m or 1 m, while the second distribution has an equal probability of outcomes of 0 m and 100 m. Both distributions have the same entropy as they assign the same probabilities to the same number of outcomes, but the first distribution has a variance of 0.5 and the second of 5000. Here, the variance has captured the discrepancy in distributions, as would be useful. For the above reasons, the variance in particle position is modelled, with additional terms representing the effect of uncertainty in ocean dynamics upon particle position, and this variance in particle position is termed the uncertainty of SCEM.

4.5.1 Describing uncertainty

This section constructs a PDE description of the uncertainty in position of a Langrangian particle ensemble, for use in sensor placement determination. In a simulation realisation denoted by s_n with $n \in \mathbb{N}$, for a selected oil particle p_i, with $i \in \mathbb{N}$, a particle has the 2D position denoted $\vec{O}(p_i, t, s_n)$ where $\vec{O} : \mathbb{N} \times \mathbb{R}_+ \times \mathbb{N} \to \mathbb{R}^2$ and is modelled as a vector valued random process. In

a slight abuse of notation, dependency upon the simulation realisation is omitted from here-on, as the uncertainty is to be defined for a single simulation realisation. For a 2D oil particle position, depth in water is ignored. Particle 2D movement is also modelled as a vector valued random process, described by

$$\frac{d\vec{O}}{dt} = f(\vec{U}_c) + f(\vec{U}_w) + f(\vec{U}_{\text{wave}}) + \vec{U}_{\text{d}} + \vec{U}_s + \vec{U}_{\text{mech}}, \tag{4.80}$$

where the ocean current velocity is $\vec{U}_c$, with $\vec{U}_w$ as the wind velocity, the vector $\vec{U}_{\text{d}}$ is the horizontal turbulent diffusion correction velocity, $\vec{U}_{\text{wave}}$ is the Stokes drift wave induced velocity. The horizontal diffusion velocity is $\vec{U}_s$ and the mechanical spreading velocity of the particle is denoted $\vec{U}_{\text{mech}}$. The right hand terms in (4.80) are all random variables and therefore the expectation (4.82) is the total expectation. The velocities $\vec{U}_c, \vec{U}_w, \vec{U}_{\text{wave}}, \vec{U}_{\text{d}}$ are independent from the presence of other oil particles and are calculated within the fluid model. The min-max normalised probability of oil presence in a cell (x_i, y_j), at a given time, is defined by $P_{\hat{\vec{O}}_p}(x_i, y_j)$ where $P_{\hat{\vec{O}}_p} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \to [0, 1]$, detailed in Hodgson et al. 2019. This indicates likely locations of oil for use in the optimisation problem of (5.1a), scaled between 0 for least probable and 1 for most probable. The min-max normalised value of the ith element $\alpha_i \in \mathbb{R}$ within the vector $\vec{\alpha} \in \mathbb{R}^{n_\alpha}$ is $\beta(\alpha_i) : \mathbb{R}^{n_\alpha} \times \mathbb{R} \to \mathbb{R}$ described by,

$$\beta(\alpha_i) = \frac{\alpha_i - \min_j(\alpha_j \in \vec{\alpha})}{\max_j(\vec{\alpha}_j \in \vec{\alpha}) - \min_j(\vec{\alpha}_j \in \vec{\alpha})}, \tag{4.81}$$

where $\min_j(\vec{\alpha}_j \in \vec{\alpha}) : \mathbb{R}^{n_\alpha} \to \mathbb{R}$ is the minimum value of $\vec{\alpha}$ and $\max_j(\vec{\alpha}_j \in \vec{\alpha}) : \mathbb{R}^{n_\alpha} \to \mathbb{R}$ is the maximum value of α. In this manner, for all $i \in [\![1, n_\alpha]\!]$, the min-max normalised vector $\vec{\beta} \in \mathbb{R}^{n_\alpha}$ can be formed by $\vec{\beta} = [\beta(\alpha_1), \beta(\alpha_2), \beta(\alpha_i), ..., \beta(\alpha_{n_\alpha})]$.

The uncertainty in particle position is the squared area in which the particle has a high probability of being in, after drifting from a known location. From the description of motion in (4.80), a single expected drift velocity $\vec{U} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^2$ is calculated for application to the oil particle by,

$$\mathbb{E}\left[\frac{d\vec{O}}{dt}\right] = \vec{U} = \mathbb{E}\left[f(\vec{U}_c)\right] + \mathbb{E}\left[f(\vec{U}_w)\right] + \mathbb{E}\left[f(\vec{U}_{\text{wave}})\right] + \mathbb{E}\left[\vec{U}_{\text{d}}\right] + \mathbb{E}\left[\vec{U}_s\right] + \mathbb{E}\left[\vec{U}_{\text{mech}}\right]. \tag{4.82}$$

The drift velocity $\vec{U}$ has a horizontal component $u : \delta\Omega \times \mathbb{R}_+ \to \mathbb{R}$ and vertical component $v : \delta\Omega \times \mathbb{R}_+ \to \mathbb{R}$. Note that for the horizontal turbulent diffusion velocity it holds that $\mathbb{E}\left[\vec{U}_s\right] = 0$. Mechanical spreading is negligible after the oil has reached a terminal thickness and therefore is discarded.

The uncertainty tracer $q_{p_i} : \mathbb{N} \times \mathbb{R}_+ \to [0, 1]$ is defined as the square of the area in which the position of a particle has probability ζ to be within, when moved from a previously known position in $\partial\Omega$ over a given time-step. The uncertainty tracer q_{p_i} is normalised to the spatial domain $\partial\Omega$ and has a minimum value of 0 corresponding to a known particle position, and a maximum value of 1 denoting that a particle can be anywhere in the spatial domain. The probable region $\mathscr{A}(p_i, t)$ is described by

$$\mathscr{A}(p_i, t) = \left\{(x, y) \in \mathbb{R}^2 : \mathbb{P}\left[\vec{O}(p_i, t + dt) \in \text{Conv}(\mathscr{A})\right] \geq \zeta\right\}, \tag{4.83}$$

where $\text{Conv}(\mathscr{A})$ is the convex hull of the set of points in $\mathscr{A}$, where $\mathscr{A}$ is defined such that this convex hull is the region that contains the particle p_i in the next-time step with probability at least ζ. Assuming the vector $\frac{d\vec{O}}{dt}$ is formed of independent, normally distributed horizontal and vertical components, the scalar area $A(p_i, t)$ of $\text{Conv}(\mathscr{A})$ can be described without needing to determine $\mathscr{A}$, where $A : \mathbb{N} \times \mathbb{R}_+ \to \mathbb{R}_+$. Modelling particle movement as a Chi-squared distribution with 2 degrees of freedom, $A(p_i, t)$ is described by

$$A(p_i, t) = \pi\chi(\zeta, 2)\sqrt{\text{Var}_x\left(\frac{d\vec{O}}{dt}\right)}\sqrt{\text{Var}_y\left(\frac{d\vec{O}}{dt}\right)}, \tag{4.84}$$

where $\chi := \chi(\zeta, 2) : [0, 1] \times \mathbb{N} \to \mathbb{R}_+$ is the Chi-squared distribution value for probability ζ with 2 degrees of freedom. In (4.84), the terms $\text{Var}_x\left(\frac{d\vec{O}}{dt}\right)$ and $\text{Var}_y\left(\frac{d\vec{O}}{dt}\right)$, where $\text{Var}_x : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \to [0, \frac{n_x \delta x}{\pi\chi}^2]$ and $\text{Var}_y : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \to [0, \frac{n_y \delta y}{\pi\chi}^2]$, are the variances of particle movement in the x-direction and y-direction respectively. Figure 4.13 is a pictorial representation. The upper bounds $\frac{n_x \delta x}{\pi\chi}^2$ and $\frac{n_y \delta y}{\pi\chi}^2$ limit the axes of the confidence interval ellipse to the width and height of the domain respectively.

Note that $\text{Var}_x\left(\frac{d\vec{O}}{dt}\right)$ and $\text{Var}_y\left(\frac{d\vec{O}}{dt}\right)$ are mappings from the spatial domain surface, time and the state vector, but not the particle index p_i. The variance of the vector valued random process describing $\frac{d\vec{O}}{dt}$ is not particle dependent, but location dependent. Given the focus

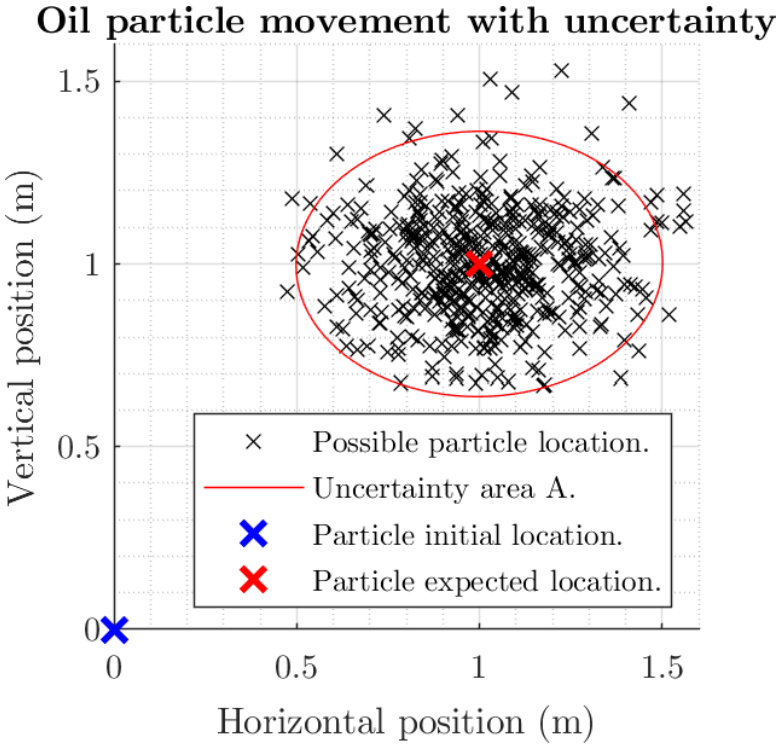


Figure 4.13: This figure displays a particle moving from the origin (blue cross) with a sampled velocity. The expected location given an expected velocity (4.82) is displayed as a red cross. Possible particle locations are marked with black crosses, with the perimeter of the chi-squared approximation of (4.83) displayed as a red ellipse, with an area described by (4.84). This is the region that contains the particle with some probability, e.g 95% probability meaning 5% of possible locations are outside the region. In practice, 99.95% is used. This red ellipse is constructed for a particle at the centre of every grid-cell, using the expected velocity of the particle in the grid cell to locate (or advect) the ellipse and the variances in particle velocity in the grid-cell to determine the ellipse axis lengths. The definition of uncertainty in (4.85) is a function of the area of the red ellipse. The PDEs in (4.86) describe the spatio-temporal evolution of the variances determining the ellipse axis lengths.

of this work upon highly mobile sensors measuring at the water surface, determination of $A(p_i, t)$ is constrained to horizontal processes for an oil particle at the water surface. Hence the random processes of particle entrainment and vertical turbulent diffusion are ignored, enabling the description of variance in (4.86). If a depth capable sensing strategy were required, $A(p_i, t)$ would be replaced by a scalar volume from a Chi-squared distribution with 3 degrees of freedom in particle movement and (4.86) would be extended by a further PDE describing the evolving variance of particle movement in the z-direction.

To recognise the dependency of uncertainty upon sensor positions and other components of the monitoring framework, now expand the definition of uncertainty $q_{p_i} : \mathbb{N} \times \mathbb{R}_+ \to [0, 1]$ as a function of the particle index and time, to $q(\overset{u}{\vec{x}}, t, \vec{p})$, where $q : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_u} \times \mathbb{R}^{1\times 2p_n} \to [0, 1]$. In this definition, $\overset{u}{\vec{x}} : \mathbb{R}_+ \to \mathbb{R}^{n_u}$ is the $n_u = 8n_x n_y$ length state-vector of the uncertainty system (see Section 4.6). Although some states in $\overset{u}{\vec{x}}$ have further dependency upon the SCEM state vector $\vec{x}$, this layered dependency is omitted for brevity. Hence q is a function of the spatial location, the time, the uncertainty state vector and $n_p \in \mathbb{N}$ sensor positions in the row vector $\vec{p} : \mathbb{R}_+ \to \mathbb{R}^{1\times 2n_p}$. A row vector is required for the optimisation solver method in Section 5.1.1. Let $k_\chi \coloneqq \frac{1}{\partial\Omega_A}\pi\chi$, $\sigma_x^2 \coloneqq \mathrm{Var}_x\left(\frac{d\vec{O}}{dt}\right)$ and $\sigma_y^2 \coloneqq \mathrm{Var}_y\left(\frac{d\vec{O}}{dt}\right)$, then the uncertainty tracer function $q(\overset{u}{\vec{x}}, t, \vec{p})$ is defined by

$$q(\overset{u}{\vec{x}}, t, \vec{p}) \coloneqq k_\chi^2 \sigma_x^2 \sigma_y^2. \tag{4.85}$$

In summary, q is a scalar measure of the area in which an oil particle might be, as defined by the Chi-squared distribution for 2 degrees of freedom, normalised to the spatial domain. The uncertainty $q(\overset{u}{\vec{x}}, t, \vec{p})$ is a function of the variance in the horizontal and vertical directions, which are modelled by the PDEs,

$$\frac{\partial\sigma_x^2}{\partial t} = -u \cdot \nabla_x \sigma_x^2 - v \cdot \nabla_y \sigma_x^2 + \nu\nabla^2\sigma_x^2 - \underline{H}_t(\vec{p}_{t-\delta t})\frac{k_s\sigma_x^2}{\delta t}\underline{H}_r(\vec{p}) + \frac{d_h(u,v)}{\delta t} + \epsilon_x + E_{\mathrm{KF}_x}(\vec{p}), \tag{4.86a}$$

and

$$\frac{\partial \sigma_y^2}{\partial t} = -u \cdot \nabla_x \sigma_y^2 - v \cdot \nabla_y \sigma_y^2 + \nu \nabla^2 \sigma_y^2 \\ - \underline{H}_t(\vec{p}_{t-\delta t}) \frac{k_s \sigma_y^2}{\partial t} \underline{H}_r(\vec{p}_{t-\delta t}) + \frac{d_h(u,v)}{\delta t} + \epsilon_y + E_{\mathrm{KF}_y}(\vec{p}). \quad (4.86b)$$

The PDEs of (4.86) are a Fokker-Planck or Kolmogrov type equation, that together with (4.85), describe the spatio-temporal evolution of the uncertainty (as a scalar area) in the position of Langrangian oil particles, across the domain. The division by the time-step scales out the time-step dependency of the PDE discretisation. These PDEs have initial values at t_0 of $\sigma_x^2 = \frac{\delta x}{\pi\chi}^2$ and $\sigma_y^2 = \frac{\delta y}{\pi\chi}^2$, meaning a uniform spatial distribution of uncertainty where particles are assumed to be anywhere within a grid-cell. In a scenario with no prior information, they have initial values of $\sigma_x^2 = \frac{n_x \delta x}{\pi\chi}^2$ and $\sigma_y^2 = \frac{n_y \delta y}{\pi\chi}^2$ meaning particles can be anywhere in the domain. In (4.86), $\nu \in \mathbb{R}_+$ is the diffusion coefficient of the fluid moving the particle, and the terms $\underline{H}_t(\vec{p}_{t-\partial t})$ and $\underline{H}_r(\vec{p})$ are used to activate sensing after sufficient time for a sensor travelling at speed $v_{\text{sensor}} \in \mathbb{R}_+$ to reach a location and to remove the uncertainty tracer in a radius $r \in \mathbb{R}_+$ around the sensor position. These $\underline{H}$ terms are functions of $H(\cdot)$, the Heaviside step function. The sensor movement term $\underline{H}_t : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^{1\times 2n_p} \to [0,1]$ has dependency upon the spatial location, time, sensor velocity and the sensor positions. The sensor reach, $\underline{H}_r : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}_+ \times \mathbb{R}^{1\times 2n_p} \to [0,1]$ has dependency upon the spatial location, time, sensor radius and the sensor positions. In (4.86), the sensor effectiveness coefficient $k_s \in [0,1]$ defines how much uncertainty as a proportion of the amount present should be removed by a reading. Here a value of 1 has been used, as perfect sensors are assumed. The variance of the random walk that models turbulent diffusion (Hodgson, Esnaola, and Jones 2019) is described by $\frac{d_h(u,v)}{\partial t} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$. Additional inputs of uncertainty in x and y directions, are $E_{\mathrm{KF}_x}(\vec{p})$ and $E_{\mathrm{KF}_y}(\vec{p})$, where $E_{\mathrm{KF}_x} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}_+$ and $E_{\mathrm{KF}_y} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}_+$. These uncertainty inputs are a function of the data assimilation error, see Section 6.1.1. The terms $\epsilon_x, \epsilon_y \in \mathbb{R}_+$ are the error covariance (or mean squared error) of $\vec{U}$ from external data sources of $\vec{U}_c$ and $\vec{U}_w$, or the sample variance of $\vec{U}$ for that spatio-temporal location in the absence of other information. Sensor $i \in \mathbb{N}$ is selected by the notation $\vec{p}_{i_t}$ where $\vec{p}_{i_t} : \mathbb{R}_+ \times \mathbb{N} \to \mathbb{R}^{1\times 2}$ returns the position vector of the sensor at the

noted time. To describe the limited Heaviside functions $\underline{H}_t$ and $\underline{H}_r$, first define the functions

$$H_t(\vec{p}_{t-\delta t}) = \sum_{i=1}^{n_p} \left(H \left(\partial t - \frac{\left\| \partial\Omega_p - \vec{p}_{i_{t-\delta t}} \right\|_2}{v_{\text{sensor}}} \right) \right), \tag{4.87}$$

and

$$H_r(\vec{p}) = \sum_{i=1}^{n_p} H(r - \| \partial\Omega_p - \vec{p}_{i_t} \|_2). \tag{4.88}$$

The term $\partial\Omega_p \in \mathbb{R}^2$ is the position in $\partial\Omega$ where the containing function is evaluated. The limited Heaviside functions are then described by,

$$\underline{H}_t(\vec{p}_{t-\delta t}) = \begin{cases} 1 & H_t(\vec{p}_{t-\delta t}) \geq 1 \\ H_t(\vec{p}_{t-\delta t}) & \text{otherwise,} \end{cases} \tag{4.89}$$

and

$$\underline{H}_r(\vec{p}) = \begin{cases} 1 & H_r(\vec{p}) \geq 1 \\ H_r(\vec{p}) & \text{otherwise,} \end{cases} \tag{4.90}$$

where the limits are necessary to describe the ineffectiveness of multiple sensors being active in the same location.

The definition of uncertainty, in (4.85) and (4.86), incorporates uncertainty propagating through the system via advection and diffusion and the additional uncertainty for oil particles due to turbulent advection and poor external data as uncertainty source parameters. There is a further uncertainty input as a function of the data assimilation error and finally a term for the removal of uncertainty through mobile sensors. The uncertainty definition considers the dynamics of an oil spill and estimation framework, with a continuous, determinable derivative and hence determinable sensitivity of uncertainty to sensor position, enabling gradient based optimisation. This definition of uncertainty is suitable for description as a non-linear state-space system (Ogata et al. 1995).

4.6 Uncertainty as a state space system (Stages 1 & 2)

This section describes the uncertainty definition in Section 3.4 as a non-linear state-space system, suitable for application of the adjoint method (Chapter 5, Section 5.2) to solve the

optimisation for sensor placement (Chapter 5, Section 5.1). Although conceptually straightforward, this is technically challenging due to the non-linear terms in (4.86). The PDEs in (4.86) are solved with forward Euler time-stepping, using δt from the fluid model, upon the staggered grid spatial discretisation of SCEM (cell boundary states are fluid velocities and fluid variances, cell centre states are all other states). However, unlike in SCEM, the cell boundary states are required at the cell centres: uncertainty is required to be advected by the fluid velocity and the velocity estimation error is required as an uncertainty source, at the cell centre. Cell centre values are the mean of the adjacent values on the cell boundary.

The section goes on to detail the implementation of bounded state values between upper and lower limits by the introduction of new matrices to the state-space system, rather than new terms in (4.86). Finally, the state-trajectory of the system is described. There is a summary of key insight for the reader at the end of this section, and the hurried reader may wish to finish this introductory section and then skip to the summary. The optimisation for a sensor trajectory does not necessarily share the initial t_0 and final time t_f of the oil trajectory simulation, so denote $t_\tau \in \mathbb{R}_+$ and $t_\epsilon \in \mathbb{R}_+$ as the initial and final times of the sensor trajectory, with corresponding discrete time steps $k_\tau \in \mathbb{N}$ and $k_\epsilon \in \mathbb{N}$ and with $n_\tau \in \mathbb{N}$ discrete time-steps in the interval $[t_\tau, t_\epsilon]$. For example, the sensor trajectory might be determined only for the last 2 steps of the oil trajectory simulation, then implemented, then the oil trajectory extrapolated to a further time with use of new sensor information, then the sensor trajectory determined again, in a receding horizon manner.

First define a trajectory of sensor locations $\vec{\mathscr{P}} \in \mathbb{R}^{1\times 2n_p n_\tau} : \vec{\mathscr{P}} = [\vec{p}_{k_\tau}, \vec{p}_{k_\tau+1}, \vec{p}_{k_\tau+2}, ..., \vec{p}_{k_\epsilon}]$, where $\vec{p}_k = \vec{\mathscr{P}}(\!|1, [\![((k-1)2n_p+1), (k2n_p)]\!]\rangle$ for the time index k. Define further scalar terms, $e_x, e_y, s_x, s_y : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}$, as $e_x \coloneqq E_{\mathrm{KF}_x}(\vec{p}, \vec{p}_{t-\delta t})$, $e_y \coloneqq E_{\mathrm{KF}_y}(\vec{p}, \vec{p}_{t-\delta t})$, $s_x \coloneqq \epsilon_x$ and $s_y \coloneqq \epsilon_y$. Then, in the uncertainty system, the state-vector is formed of the components $\vec{\sigma}_x^2, \vec{\sigma}_y^2, \vec{e}_x, \vec{e}_y : \mathbb{R}_+ \to \mathbb{R}^{n_x n_y}$, that are each a vector of a spatially distributed scalar value sampled at $n_x n_y$ locations and a function of time. These are the particle movement variance in the horizontal and vertical directions and the variance sources from the data assimilation in horizontal and vertical directions at cell-centres, respectively. The system input vector is formed of vectors of the cell-centre fluid velocity components $\vec{u}, \vec{v} : \mathbb{R}_+ \to \mathbb{R}^{n_x n_y}$ and further vectors $\vec{s}_x, \vec{s}_y : \mathbb{R}_+ \to \mathbb{R}^{n_x n_y}$. These are externally determined inputs of uncertainty to the system. The uncertainty system

utilises a state vector $\overset{a}{\vec{x}}_k = [\vec{\sigma}_x^2; \vec{\sigma}_y^2; \vec{e}_x; \vec{e}_y] \in \mathbb{R}^{4n_x n_y}$ and inputs $\vec{i}_k = [\vec{u}; \vec{v}; \vec{s}_x; \vec{s}_y] \in \mathbb{R}^{4n_x n_y}$. In the next section, the abstract structure of the state-space system will be described, without rigorous definition of terms. This will facilitate a more rigorous definition later.

4.6.1 Forming the abstract state space system

In this section, a traditional state space system is formed, then the state-vector is augmented with the internally determined inputs. This is because the optimisation and Adjoint method based solver require a complete system description, including inputs, in the state trajectory. Assume fixed sensor positions and that the system has no time dependency outside the current time-step k, then a non-linear (in both states and inputs) state-space system is described by

$$\dot{\overset{a}{\vec{x}}}_k = \overset{ss}{\boldsymbol{A}}_k(\overset{a}{\vec{x}}_k, \vec{i}_k)\overset{a}{\vec{x}}_k + \overset{ss}{\boldsymbol{B}}_k(\overset{a}{\vec{x}}_k, \vec{i}_k)\vec{i}_k, \tag{4.91a}$$

$$\vec{y}_k = \overset{ss}{\boldsymbol{C}}_k(\overset{a}{\vec{x}}_k, \vec{i}_k)\overset{a}{\vec{x}}_k + \overset{ss}{\boldsymbol{D}}_k(\overset{a}{\vec{x}}_k, \vec{i}_k)\vec{i}_k. \tag{4.91b}$$

In this system, $\dot{\overset{a}{\vec{x}}}_k$ is the time-derivative of the states, $\overset{ss}{\boldsymbol{A}}_k$ is the state dynamics matrix, $\overset{ss}{\boldsymbol{B}}_k$ is the input matrix, $\overset{ss}{\boldsymbol{C}}_k$ is the output or sensor matrix and $\overset{ss}{\boldsymbol{D}}_k$ is the feed-through matrix. More rigorous definitions will be provided later. Note the inputs $\vec{i}_k$ are not the sensor positions to be determined, but are sources of uncertainty prescribed externally. Hence, the system is redefined, with a new state vector $\overset{u}{\vec{x}}_k = [\overset{a}{\vec{x}}_k; \vec{i}_k]$ where $\overset{u}{\vec{x}}_k := \overset{u}{\vec{x}}_k(t) : \mathbb{R}_+ \to \mathbb{R}^{n_u}$ for $n_u = 8n_x n_y$ total states, and new matrices

$$\boldsymbol{A}_k = \begin{bmatrix} \overset{ss}{\boldsymbol{A}}_k & \overset{ss}{\boldsymbol{B}}_k \\ 0^{\overset{ss}{\boldsymbol{A}}} & 0^{\overset{ss}{\boldsymbol{B}}} \end{bmatrix} \tag{4.92}$$

and

$$\boldsymbol{C}_k = \begin{bmatrix} \overset{ss}{\boldsymbol{C}}_k & \overset{ss}{\boldsymbol{D}}_k \end{bmatrix}, \tag{4.93}$$

where the dependency notation describing non-linearity is temporarily discarded for brevity. Here the notation $0^{\overset{ss}{\boldsymbol{A}}}$ is a matrix of zeros of the same size as $\overset{ss}{\boldsymbol{A}}$ and $0^{\overset{ss}{\boldsymbol{B}}}$ is a matrix of zeros of the same size as $\overset{ss}{\boldsymbol{B}}$. The system description now utilises $\vec{p}_k$ as inputs with a corresponding input matrix $\boldsymbol{B}_k$. Now define a further trajectory, $\overset{u}{\vec{\mathscr{X}}} \in \mathbb{R}^{n_\tau n_u}$ that contains the vertically

concatenated states at each time-step, where $\overset{u}{\vec{\mathscr{X}}} = [\overset{u}{\vec{x}}_0, \overset{u}{\vec{x}}_1, \overset{u}{\vec{x}}_k, ..., \overset{u}{\vec{x}}_{n_\tau - 1}]$. To create a general description, now recognise the time-varying nature of sensor positions $\vec{p}_k$ in $\mathscr{P}$ and by examination of (4.86) the non-linearity due to dependency, both on $\vec{p}_k$ and $\vec{p}_{k-1}$. There is further non-linearity on $\vec{p}_{k+1}$, but this will be discussed later. For now, the non-linearity will be represented in a time-varying description of $\boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \mathscr{P})$, which encompasses non-linearity of the system at k due to any states or sensor positions in the trajectory. A new input matrix is also defined, $\boldsymbol{B}_k(\overset{u}{\vec{\mathscr{X}}}, \mathscr{P})$. The new state-space system, discarding the output equation, is described by

$$\dot{\overset{u}{\vec{x}}}_k = \boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})\overset{u}{\vec{x}}_k + \boldsymbol{B}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})\vec{p}_k. \tag{4.94a}$$

After inspection of (4.86), it is decided to represent the complete system in $\boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})\overset{u}{\vec{x}}_k$, as there is no summation term of $\vec{p}_k$. Hence, $\boldsymbol{B}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}}) = 0$ and control of the system states is achieved through manipulation of the system dynamics matrix.

4.6.2 Constructing the system dynamics matrix

This section describes the construction of the state space system dynamics matrix, that contains the dynamics of the uncertainty PDEs discretised form. This enables the evolution of oil particle position uncertainty to be utilised in a sensor placement optimisation. The uncertainty system inherits the grid spatial discretisation of the fluid model in SCEM but utilises cell centred velocities $\vec{u}, \vec{v} \in \mathbb{R}^{n_x n_y}$. These are the mean of the adjacent velocities that lie upon cell boundaries. An explicit forward Euler time discretisation is used, with the time-step δt from SCEM. When solving backwards in time with the adjoint method, this forms a useful implicit solver for the adjoint variable vector. In (4.86) let the operation $\vec{u}\cdot\nabla_x$ be represented by a linear algebra finite difference operation $\text{diag}(\vec{u})\boldsymbol{D}_x$, where $\boldsymbol{D}_x$ is the 1st to 3rd ordered single-sided difference for the first derivative as prescribed by boundaries, otherwise 2nd to 3rd ordered centred difference for the first derivative, matrix in the x direction. Let ∇^2 be represented by the 1st ordered centred difference matrix for the second derivative $\boldsymbol{D}^2$. Assume similar definitions of finite difference matrices in the y direction. Recall the definition of horizontal diffusivity, from Section 4.2.1, and let $c_\text{d} = \frac{c_\text{smag}}{\frac{1}{\delta x^2 + \delta y^2}}$. Define new functions $\boldsymbol{T}_k(\vec{u}_k, \vec{v}_k)_u$ and

$\boldsymbol{T}_k(\vec{u}_k, \vec{v}_k)_v$, where $\boldsymbol{T}_k : \mathbb{R}^{n_x n_y} \times \mathbb{R}^{n_x n_y} \to \mathbb{R}^{n_x n_y}$. The functions are described by

$$\boldsymbol{T}_k(\vec{u}_k, \vec{v}_k)_u = \operatorname{diag}\left(\vec{u}^{\circ -1} \circ c_{\mathrm{d}}\left((\boldsymbol{D}_x \vec{u}_k - \boldsymbol{D}_y \vec{v}_k) + (\boldsymbol{D}_y \vec{u}_k + \boldsymbol{D}_x \vec{v}_k)\right)^{\circ 0.5}\right), \tag{4.95}$$

and

$$\boldsymbol{T}_k(\vec{u}_k, \vec{v}_k)_v = \operatorname{diag}\left(\vec{v}^{\circ -1} \circ c_{\mathrm{d}}\left((\boldsymbol{D}_x \vec{u}_k - \boldsymbol{D}_y \vec{v}_k) + (\boldsymbol{D}_y \vec{u}_k + \boldsymbol{D}_x \vec{v}_k)\right)^{\circ 0.5}\right). \tag{4.96}$$

Then define the function $\boldsymbol{Y}_k(\vec{p}_k, \vec{p}_{k-1}) : \mathbb{R}^{1\times 2n_p} \times \mathbb{R}^{1\times 2n_p} \times \mathbb{R}^{n_x n_y} \times \mathbb{R}^{n_x n_y} \times \mathbb{R}_+ \to \mathbb{R}^{n_x n_y \times n_x n_y}$ as,

$$\boldsymbol{Y}_k(\vec{p}, \vec{p}_{k-1}) \coloneqq -\operatorname{diag}(u_k)\boldsymbol{D}_x - \operatorname{diag}(v_k)\boldsymbol{D}_y + \nu \boldsymbol{D}^2 - \operatorname{diag}(\underline{H}_t(\vec{p}_{k-1}))\frac{k_s}{\delta t}\operatorname{diag}(\underline{H}_r(\vec{p})). \tag{4.97}$$

Note that in $\boldsymbol{Y}_k$, the terms $\operatorname{diag}(\underline{H}_t(\vec{p}_{k-1}))$ and $\operatorname{diag}(\underline{H}_r(\vec{p}))$ form $n_x n_y \times n_x n_y$ diagonal matrices, containing the evaluation of the Heaviside term for the spatial locations in the discretised $\partial\Omega$ along the diagonal. These Heaviside terms activate sensing for travel time and a sensor radius respectively.

Then define further state transition functions $\vec{E}_{k+1_x}(\vec{p}_k, \vec{p}_{k+1}), \vec{E}_{k+1_y}(\vec{p}_k, \vec{p}_{k+1}) : \mathbb{R}^{1\times 2n_p} \to \mathbb{R}^{n_x n_y}$, that step the horizontal and vertical components of the estimated error covariance $E_{\mathrm{KF}}(\vec{p}_k, \vec{p}_{k-1})$ to the next time-step and apply the posterior Kalman filter correction. The variance sources $E_{\mathrm{KF}_x}(\vec{p})$ and $E_{\mathrm{KF}_y}(\vec{p})$ are defined as a function of the sensor positions for brevity. However, they are also a function of the reduced order Kalman filter estimated error covariance matrix $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k}$ in the RTS smoother (see Chapter 6), where for the horizontal component

$$E_{\mathrm{KF}_x}(\vec{p}, \vec{p}_{t-\delta t}) = \boldsymbol{G}_x\left(\operatorname{diag}\left(\boldsymbol{\Psi}_u \boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{t-\delta t})_{t|k} \boldsymbol{\Psi}_u^*\right) + \max_{\mathrm{row}}\left(\widetilde{\mathscr{Z}}_x\right)\right). \tag{4.98}$$

The term $\boldsymbol{G}_x \in \mathbb{R}^{n_x n_y \times (n_x+1) n_y}$ is the matrix representation of the linear operation that averages adjacent cell-boundary values, to form the cell-centred value. Define the cell-boundary horizontal velocities as $\overset{e}{\vec{u}} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^{(n_x+1)n_y}$ then $\vec{u} = \boldsymbol{G}_x \overset{e}{\vec{u}}$. The multiplication $\boldsymbol{G}_x \overset{e}{\vec{u}}$ computes the mean velocity at every cell centre, using the two adjacent cell-boundary velocities. In (4.98) the cell-boundary horizontal velocity uncertainty is averaged (using the mean of adjacent values), to produce a cell-centred horizontal velocity uncertainty, by multiplication with $\boldsymbol{G}_x$. Similarly, $\boldsymbol{G}_y \in \mathbb{R}^{n_x n_y \times n_x (n_y+1)}$ is the matrix representation of the linear operation that averages adja-

cent cell-boundary vertical velocities, to form the cell-centred vertical velocities. Define the cell-boundary vertical velocities as $\overset{e}{\vec{v}} : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^{n_x(n_y+1)}$ then $\vec{v} = \boldsymbol{G}_y \overset{e}{\vec{v}}$.

The expression for the vertical component of uncertainty is similar to (4.98), but with $\boldsymbol{G}_y$, $\boldsymbol{\Psi}_v$ and $\widetilde{\mathscr{Z}}_y$. See Chapter 6 for details on the data assimilation process and estimation error. In short, $\boldsymbol{\Psi}_u \in \mathbb{C}^{(n_x+1)n_y}$ and $\boldsymbol{\Psi}_v \in \mathbb{C}^{n_x(n_y+1)}$ are subsets of the reduced order mode matrix $\boldsymbol{\Psi}$. These subsets map the reduced order states to the current-time horizontal and vertical cell-boundary velocities respectively, and in this case, maps the reduced order estimation error covariance to the mean-squared estimation error for the horizontal and vertical cell-boundary velocities. Note that in (4.98), $\max_{\text{row}}\left(\widetilde{\mathscr{Z}}\right)$ is a measure of error between the model and the environment, and is deemed impossible to predict and so is assumed constant. The error state transition function in the horizontal x direction is described by

$$\vec{E}_{k+1_x}(\vec{p}_{k+1}, \vec{p}_k) = \boldsymbol{G}_x\Bigg(\operatorname{diag}\left(\boldsymbol{\Psi}_u\left(\left(\boldsymbol{I} - \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)\boldsymbol{C}_{\text{KF}_{k+1}}(\vec{p}_{k+1})\right)\right)\left(\Lambda^{\frac{\delta t}{\Delta t}}\boldsymbol{P}_{\text{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}(\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_e\right)\right)\boldsymbol{\Psi}_u^*\right) + \max_{\text{row}}\left(\widetilde{\mathscr{Z}}_x\right)\Bigg), \quad (4.99)$$

with a similar description for the vertical y direction, with v and y subscripts. Again, see Chapter 6 for details, but in short, $\boldsymbol{L}$ is the Kalman filter gain, $\boldsymbol{C}_{\text{KF}}$ is a sensor matrix, $\Lambda^{\frac{\delta t}{\Delta t}}$ is the reduced order state transition matrix from one time-step to the next and $\boldsymbol{R}_e$ is the reduced order model state covariance matrix. Note in (4.99) there is dependency upon $\vec{p}_{k-1}$, in $\boldsymbol{P}_{\text{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}$, which in turn has an unstated dependency upon $\vec{p}_{k-2}$ and so on, as a result of being an iterative process. Provided the adjoint description accounts for the effect of the iteration directly before and after, any implicit iterative dependency will be accounted for.

To describe the state transition of $\vec{e}_{x_k}$ and $\vec{e}_{y_k}$ when multiplied with the state-vector $\vec{x}_k$, define a new term $\boldsymbol{Q}_{k+1_x}(\vec{e}_{x_k}, \vec{p}_{k+1}, \vec{p}_k)$, to be utilised within the state transition matrix. The term $\boldsymbol{Q}_{k+1_x}(\vec{e}_{x_k}, \vec{p}_{k+1}, \vec{p}_k)$ is described by,

$$\boldsymbol{Q}_{k+1_x}(\vec{e}_{x_k}, \vec{p}_{k+1}, \vec{p}_k) = \operatorname{diag}\left(\left(\vec{e}_x^{\circ -1}\right) \circ \vec{E}_{k+1_x}(\vec{p}_{k+1}, \vec{p}_k)\right), \quad (4.100)$$

such that

$$\vec{e}_{x_{k+1}} = \boldsymbol{Q}_{k+1_x}(\vec{e}_{x_k}, \vec{p}_{k+1}, \vec{p}_k)\vec{e}_{x_k}, \tag{4.101}$$

with similar expressions for the vertical direction, with y subscripts.

Recall that $\overset{u}{\vec{x}}_k = [\overset{a}{\vec{x}}_k; \vec{i}_k]$ and define a modified identity matrix $\boldsymbol{I}^{\vec{i}}$ such that $\boldsymbol{I}^{\vec{i}}\overset{u}{\vec{x}}_k = [0^{\overset{a}{\vec{x}}_k}; \vec{i}_k]$, where $0^{\overset{a}{\vec{x}}_k}$ is a vector of n_u zeros. Under forward Euler stepping, the state transition is then described by

$$(\boldsymbol{I} - \boldsymbol{I}^{\vec{i}})\overset{u}{\vec{x}}_{k+1} + \boldsymbol{I}^{\vec{i}}\overset{u}{\vec{x}}_k = \boldsymbol{I}\overset{u}{\vec{x}}_k + \delta t \boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})\overset{u}{\vec{x}}_k. \tag{4.102}$$

In (4.102) the left-hand-side terms reflect that the system states, $\overset{a}{\vec{x}}_k \subset \overset{u}{\vec{x}}_k$ have changed to $\overset{a}{\vec{x}}_{k+1}$, while the external inputs $\vec{i}_k \subset \overset{u}{\vec{x}}_k$ have remained $\vec{i}_k$.

The state dynamics matrix $\boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})$ is described by,

$$\boldsymbol{A}_k(\overset{u}{\vec{\mathscr{X}}}, \vec{\mathscr{P}}) = \begin{bmatrix} \boldsymbol{Y}_k(\cdot) & 0 & \boldsymbol{I} & 0 & \boldsymbol{T}_k(\cdot)_u & 0 & \boldsymbol{I} & 0 \\ 0 & \boldsymbol{Y}_k(\cdot) & 0 & \boldsymbol{I} & 0 & \boldsymbol{T}_k(\cdot)_v & 0 & \boldsymbol{I} \\ 0 & 0 & \boldsymbol{Q}_{k+1_x}(\cdot) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \boldsymbol{Q}_{k+1_y}(\cdot) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.103}$$

4.6.3 Describing the system trajectory and introducing limits

As noted in Section 4.5.1, there is an upper and lower bound on the state values of $\vec{\sigma}_x^2$ and $\vec{\sigma}_y^2$. Without modification, the system in (4.102) using forward Euler time discretisation undesirably steps to unlimited states and unmodified inputs. Let $\overset{u}{\underset{\sim}{\vec{x}}} \in \mathbb{R}^{n_u}$ and $\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}} \in \mathbb{R}^{n_\tau n_u}$ be the bounded state vector and trajectory of bounded states respectively. The state transition in (4.102) is now described by,

$$(\boldsymbol{I} - \boldsymbol{I}^{\vec{i}})\overset{u}{\vec{x}}_{k+1} + \boldsymbol{I}^{\vec{i}}\overset{u}{\underset{\sim}{\vec{x}}}_k = \boldsymbol{I}\overset{u}{\vec{x}}_k + \delta t \boldsymbol{A}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})\overset{u}{\underset{\sim}{\vec{x}}}_k. \tag{4.104}$$

The left-hand-side terms describe the stepping of the state vector forward, to $k+1$, but without bounded values or new inputs at $k+1$. Using 2 steps for illustration, rather than n_τ steps, the

state trajectory of this system is described by

$$\begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ -\left(\boldsymbol{I} + \delta t \boldsymbol{A}_0(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})\right) & \boldsymbol{I} & 0 \\ 0 & -\left(\boldsymbol{I} + \delta t \boldsymbol{A}_1(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})\right) & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \overset{u}{\underset{\sim}{\vec{x}}}_0 \\ \overset{u}{\underset{\sim}{\vec{x}}}_1 \\ \overset{u}{\underset{\sim}{\vec{x}}}_2 \end{bmatrix} = \begin{bmatrix} \overset{u}{\underset{\sim}{\vec{x}}}_0 \\ (\overset{u}{\underset{\sim}{\vec{x}}}_1 - \overset{u}{\underset{\sim}{\vec{x}}}_1) + (\vec{u}_1 - \vec{u}_0) \\ (\overset{u}{\underset{\sim}{\vec{x}}}_2 - \overset{u}{\underset{\sim}{\vec{x}}}_2) + (\vec{u}_2 - \vec{u}_1) \end{bmatrix}, \tag{4.105}$$

where

$$\begin{bmatrix} \overset{u}{\underset{\sim}{\vec{x}}}_0 \\ (\overset{u}{\underset{\sim}{\vec{x}}}_1 - \overset{u}{\underset{\sim}{\vec{x}}}_1) + (\vec{u}_1 - \vec{u}_0) \\ (\overset{u}{\underset{\sim}{\vec{x}}}_2 - \overset{u}{\underset{\sim}{\vec{x}}}_2) + (\vec{u}_2 - \vec{u}_1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ -\boldsymbol{I}^{\vec{i}} & \boldsymbol{I}^{\vec{i}} & 0 \\ 0 & -\boldsymbol{I}^{\vec{i}} & \boldsymbol{I}^{\vec{i}} \end{bmatrix} \begin{bmatrix} \overset{u}{\underset{\sim}{\vec{x}}}_0 \\ \overset{u}{\underset{\sim}{\vec{x}}}_1 \\ \overset{u}{\underset{\sim}{\vec{x}}}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ (\overset{u}{\underset{\sim}{\vec{x}}}_1 - \overset{u}{\underset{\sim}{\vec{x}}}_1) \\ (\overset{u}{\underset{\sim}{\vec{x}}}_2 - \overset{u}{\underset{\sim}{\vec{x}}}_2) \end{bmatrix} \tag{4.106}$$

To describe the limiting action in a general framework, new matrices external to the underlying equations in (4.86) are introduced to the state-space description. First define a state transition function $\vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) : \mathbb{R}^{n_u} \times \mathbb{R}^{1\times 2n_p} \times \mathbb{R}^{n_\tau n_u} \times \mathbb{R}^{2n_p n_\tau} \to \mathbb{R}^{n_u}$ by,

$$\vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) \coloneqq \boldsymbol{I}\overset{u}{\underset{\sim}{\vec{x}}}_k + \delta t \boldsymbol{A}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})\overset{u}{\underset{\sim}{\vec{x}}}_k. \tag{4.107}$$

To ensure the use of bounded state vectors on the system trajectory, define further matrices $\underset{\sim}{\boldsymbol{M}}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) : \mathbb{R}^{n_u} \times \mathbb{R}^{1\times 2n_p} \times \mathbb{R}^{n_\tau n_u} \times \mathbb{R}^{2n_p n_\tau} \to \mathbb{R}^{n_u \times n_u}$ and $\underset{\sim}{\boldsymbol{R}}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) : \mathbb{R}^{n_u} \times \mathbb{R}^{1\times 2n_p} \times \mathbb{R}^{n_\tau n_u} \times \mathbb{R}^{2n_p n_\tau} \to \mathbb{R}^{n_u \times n_u}$. Though explicitly noted with the bounded state and position trajectories as arguments, intuitively these functions have dependency upon the next state-vector and then inherit any dependencies of the state transition function $\vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})$.

$$\text{Define } \underset{\sim}{\boldsymbol{M}}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) \text{ such that: } \underset{\sim}{\boldsymbol{M}}_k\vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) = \overset{u}{\underset{\sim}{\vec{x}}}_{k+1} \tag{4.108a}$$

$$\text{Then: } \underset{\sim}{\boldsymbol{M}}_k(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) = \operatorname{diag}\left(\overset{u}{\underset{\sim}{\vec{x}}}_{k+1} \circ \vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}})^{\circ -1}\right), \tag{4.108b}$$

where $\operatorname{diag}(\cdot)$ forms a matrix with the operated values on the diagonal. Note how $\underset{\sim}{\boldsymbol{M}}_k$ is defined for zero valued limits but becomes undefined if it attempts to operate on a zero valued state. Thus the need for an alternative approach on the right hand side:

$$\text{Define } \underset{\sim}{\boldsymbol{R}}_{k+1}(\overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) \text{ such that: } \underset{\sim}{\boldsymbol{R}}_{k+1}\overset{u}{\underset{\sim}{\vec{x}}}_{k+1} = \overset{u}{\underset{\sim}{\vec{x}}}_{k+1} - \vec{\mathscr{G}}(\overset{u}{\underset{\sim}{\vec{x}}}_k, \vec{p}_k, \overset{u}{\underset{\sim}{\vec{\mathscr{X}}}}, \vec{\mathscr{P}}) \tag{4.109a}$$

$$\text{Then: } \boldsymbol{R}_{k+1}(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) = \operatorname{diag}\left(1 - \vec{\mathscr{G}}(\overset{u}{\vec{\underline{x}}}_k, \vec{p}_k, \overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) \circ \left(\overset{u}{\vec{\underline{x}}}_{k+1}\right)^{\circ -1}\right). \tag{4.109b}$$

Though only evaluated if $\vec{\mathscr{G}}(\overset{u}{\vec{\underline{x}}}_k, \vec{p}_k, \overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) = 0$, this full expression is needed to describe the derivative for the adjoint method. The right-hand-side of (4.106) is now described by

$$\begin{bmatrix} \overset{u}{\vec{\underline{x}}}_0 \\ (\overset{u}{\vec{\underline{x}}}_1 - \overset{u}{\vec{\underline{x}}}_1) + (\vec{u}_1 - \vec{u}_0) \\ (\overset{u}{\vec{\underline{x}}}_2 - \overset{u}{\vec{\underline{x}}}_2) + (\vec{u}_2 - \vec{u}_1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & 0 & 0 \\ -\boldsymbol{I}^{\vec{i}} & \boldsymbol{I}^{\vec{i}} & 0 \\ 0 & -\boldsymbol{I}^{\vec{i}} & \boldsymbol{I}^{\vec{i}} \end{bmatrix} \begin{bmatrix} \overset{u}{\vec{\underline{x}}}_0 \\ \overset{u}{\vec{\underline{x}}}_1 \\ \overset{u}{\vec{\underline{x}}}_2 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & \underline{\boldsymbol{R}}_1 & 0 \\ 0 & 0 & \underline{\boldsymbol{R}}_2 \end{bmatrix} \begin{bmatrix} \overset{u}{\vec{\underline{x}}}_0 \\ \overset{u}{\vec{\underline{x}}}_1 \\ \overset{u}{\vec{\underline{x}}}_2 \end{bmatrix}. \tag{4.110}$$

The trajectory of the system is then described by

$$\underbrace{\begin{bmatrix} 0 & 0 & 0 \\ -\underline{\boldsymbol{M}}_0\left(\boldsymbol{I} + \delta t \boldsymbol{A}_0\right) + \boldsymbol{I}^{\vec{i}} & \boldsymbol{I} - \left(\boldsymbol{I}^{\vec{i}} + \underline{\boldsymbol{R}}_1\right) & 0 \\ 0 & -\underline{\boldsymbol{M}}_1\left(\boldsymbol{I} + \delta t \boldsymbol{A}_1\right) + \boldsymbol{I}^{\vec{i}} & \boldsymbol{I} - \left(\boldsymbol{I}^{\vec{i}} + \underline{\boldsymbol{R}}_2\right) \end{bmatrix} \begin{bmatrix} \overset{u}{\vec{\underline{x}}}_0 \\ \overset{u}{\vec{\underline{x}}}_1 \\ \overset{u}{\vec{\underline{x}}}_2 \end{bmatrix}}_{\boldsymbol{F}(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}})} = 0. \tag{4.111}$$

In (4.111) the dependency notation has been omitted for brevity. Define the left-hand-side of the equation, in matrix form, as $\boldsymbol{F}(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) : \mathbb{R}^{n_\tau n_u} \times \mathbb{R}^{1 \times 2 n_p n_\tau} \to \mathbb{R}^{n_\tau n_u \times n_\tau n_u}$. From the trajectory description in (4.111), a Differential-Algebraic Equation is described by

$$\begin{aligned} \mathscr{F}(\overset{u}{\vec{\underline{x}}}_{k+1}, \overset{u}{\vec{\underline{x}}}_k, \vec{p}_k, \overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) = &\underbrace{\left[-\underline{\boldsymbol{M}}_k(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}})\left(\boldsymbol{I} + \delta t \boldsymbol{A}_k(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}})\right) + \boldsymbol{I}_k^i\right] \overset{u}{\vec{\underline{x}}}_k}_{\mathscr{A}_k} \\ &\underbrace{+ \left[\left(\boldsymbol{I} - \left(\boldsymbol{I}_{k+1}^i + \underline{\boldsymbol{R}}_{k+1}(\overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}})\right)\right)\right] \overset{u}{\vec{\underline{x}}}_{k+1}}_{\mathscr{C}_{k+1}} = 0, \end{aligned} \tag{4.112}$$

where $\mathscr{A}_k$ and $\mathscr{C}_{k+1}$ define terms for derivatives for the adjoint method, in Appendix A. Hence there is a second definition of $\boldsymbol{F}$, described by the stacking of the DAEs to form a trajectory,

$$\begin{bmatrix} 0 \\ \mathscr{F}(\overset{u}{\vec{\underline{x}}}_1, \overset{u}{\vec{\underline{x}}}_0, \vec{p}_0, \overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) \\ \mathscr{F}(\overset{u}{\vec{\underline{x}}}_2, \overset{u}{\vec{\underline{x}}}_1, \vec{p}_1, \overset{u}{\vec{\underline{\mathscr{X}}}}, \vec{\mathscr{P}}) \end{bmatrix} = \boldsymbol{F}. \tag{4.113}$$

It is this trajectory that describes the evolution of the uncertainty system for implementation of the Adjoint method. Note that the condition of each DAE being equal to zero still holds.

4.6.4 Uncertainty as a state space system: Key insight

The purpose of Section 4.5 is to form a description of uncertainty and sensors suitable for use in the optimisation and adjoint method of solving the sensor pathing optimisation, presented in Chapter 5. Note that this state-space system is highly non-linear, in both states and sensor positions, but it is a deterministic description of the variance in the stochastic system of Section 4.1. The bounded values for the variances in Section 4.5.1 are enforced in the state-space system through introduction of limiting matrices $\underline{\boldsymbol{M}}$ and $\underline{\boldsymbol{R}}$. The state-space system trajectory is contained in $\boldsymbol{F}$ and a Differential-Algebraic Equation for each time-step is $\mathscr{F}$; these will be heavily utilised in the adjoint based optimisation solver.

Result: Forwards ocean and contaminant simulation with external/sensor data
`/* INITIALISE */`
read user parameters (domain bounds, empirical parameters, see Table 4.1);
load external data files (domain bathymetry, external flow forcing data);
initialise domain, initialise SWEM wave models, set start date state values;
load time-varying contaminant source file (oil type, source location, leak rate);
`/* RUN SIMULATION */`
while *start time* $\leq$ *current time* $\leq$ *end time* **do**
  `/* CURRENT STATES */`
  get predicted state values;
  get external and/or sensor state values;
  correct state values using external and/or sensor values;
  calculate ekman wind value for each grid cell (4.40);
  save corrected state values;
  `/* CURRENT OIL SPILL */`
  get oil particles;
  get external and/or sensor oil values;
  correct oil spill particles using external and/or sensor values;
  save oil particles;
  `/* PREDICT NEXT OIL SPILL */`
  get corrected state values and oil particles;
  calculate time-step;
  add source oil particles for time-step;
  calculate velocity profiles at oil containing grid cells (4.36), (4.39), (4.55);
  simulate SWEM at oil containing grid cells;
  calculate oil diffusion coefficient at oil containing grid cells (4.50a)(4.50d);
  calculate total oil velocity profile at oil containing grid cells (4.46);
  calculate diffusion correction velocity at oil containing grid cells (4.47);
  foreach *oil particle* **do**
    if *oil particle is entrained into water column* (4.58) **then**
      insert oil particle at calculated depth, set buoyancy to 0 for time-step;
    end
    surface oil particles in water column if they would reach the surface by buoyancy velocity alone ;
    advect oil particle by current time local (total oil velocity + diffusion velocities + correction velocity + buoyancy velocity) for time-step (4.46);
  end
  calculate oil spill thickness and volume for each grid cell (4.60);
  foreach *oil particle* **do**
    advect oil particle by local mechanical spreading (4.62);
  end
  calculate oil spill thickness and volume for each grid cell (4.60);
  increase oil particles age;
  save predicted oil particles for next time;
  `/* PREDICT NEXT STATES */`
  simulate ocean and wind flow for time-step (4.1a);
  save predicted state values for next time;
  `/* STEP TIME */`
  step forward current time by time-step;
end

Algorithm 1: Pseudocode of the fluid model, simulating forwards in time.

America Grande simulation parameters				
Symbol	Description	Value	Nominal distribution within bounds	Units
α_{w_0}	Oil wind advection coefficient	0.02	$\alpha_{\mathrm{w}_0} \sim \mathrm{U}(0.005, 0.03)$	-
α_{c_0}	Oil current advection coefficient	1.0	$\alpha_{\mathrm{c}_0} \sim \mathrm{U}(0.9, 1.1)$	-
c_{smag}	Diffusion empirical parameter	0.1	$c_{\mathrm{smag}} \sim \mathrm{U}(0.01, 0.3)$	-
t_0	Spill leak start date	March 12th 2019 - 03:30	$t_0 \sim$ U(11/03/19 - 22:00, 12/03/19 - 17:00)	-
t_f	Spill leak end date	March 12th 2019 - 16:30	$t_f \sim$ U(12/03/19 - 12:00, 12/03/19 - 18:00)	-
V_T	Total spill volume	2200	$V_T \sim \mathrm{U}(10, 2200)$	Tonnes
n_x, n_y, n_z	Domain nodes	$64 \times 42 \times 200$	-	-
Ω	Domain size	664.3×443.0	-	km
-	Oil Properties	Heavy Fuel Oil (No. 6 Fuel oil)	-	-

Table 4.1: Tabulation of parameters used in the Grande America oil spill simulation. The notation $U(a, b)$ refers to a uniform distribution between the values a, b inclusive.

Chapter 5

Sensor Guidance and Optimisation (stage 2)

This chapter describes the sensor guidance stage of the monitoring framework; detailing the utilisation of the modelling work and posing of an optimisation, the solution to which is an optimal sensing path. The optimisation posed is to minimise the uncertainty in important oil properties for an oil spill over a receding horizon spatio-temporal domain, by guiding mobile sensors. Sensors measure the oil thickness and wind and ocean flow velocities at locations that best inform the model of the oil spill, and crucially, inform the model of environmental information that can be used to correct the model future and past extrapolations.

The approach here is akin to the uncertainty minimisation methods discussed in Chapter 2, with certain problem specific differences and extensions. In Zammit Mangion, Anderson, and Kadirkamanathan 2011, sensor constraints are imposed through sub-regioning, the one-step-ahead optimisation requires a two-step-ahead prediction, with a cost function that drives exploitation and exploration. The cost function also contains a sensor effort term. Here, sensor constraints are imposed through a Lagrangian constraint term in the cost function and an inequality constraint on the solver. A one-step-ahead optimisation and prediction is feasible with Heaviside functions describing sensor movement and range within a single step. The cost function disregards sensor effort and combines information utility measures (Hutchinson, Liu, and Chen 2019) and variance minimisation (Zammit Mangion, Anderson, and Kadirkamanathan 2011). This is similar to the Expected Improvement measure of Efficient Global Optimisation, which sums a term describing the improvement in optimisation with a term describing the

uncertainty of the surrogate model. Here, the constant spatial weighting matrix in Zammit Mangion, Anderson, and Kadirkamanathan 2011 is replaced by a spatio-temporally varying matrix. This matrix weights the reduction of uncertainty in a region, by the probability of useful information in that region and the estimated value of information available in that region, at a given time. The optimisation in this section is solved through application of the Adjoint method to determine a gradient for gradient based methods, described for one-step-ahead sensor placement and a complete Model-Predictive-Control style N-step-ahead sensor trajectory calculation which may be infeasible for the computational resources available on a laptop or UAV.

5.1 Problem Description

This section describes the optimisation used by the monitoring frame-work for a trajectory of sensor positions between t_τ and t_ϵ and then for a single vector of sensor positions between $t_\iota \in \mathbb{R}_+$ and $t_\phi \in \mathbb{R}_+$.

The optimisation cost function J, is defined in (5.1a). The optimisation seeks sensor positions $\vec{p}$ that minimise the uncertainty tracer q and constraint function c, weighted over time and space by E. These terms are formally defined below. The optimisation is subject to constraints on tracer dynamics (5.1b), tracer variance dynamics in (5.1c) and (5.1d), oil and fluid dynamics (5.1e) to (5.1i) and sensor placement (5.1j). Note that in the absence of any oil information, the optimisation is akin to the minimisation of the estimated error covariance of the ocean and

wind flow over a spatio-temporal domain. The optimisation is formulated as follows,

$$\underset{\vec{\mathscr{P}}}{\text{minimize}} \quad J = \int_{t_\tau}^{t_\epsilon} \int_{\partial\Omega} \left(E(\vec{x}, t, \vec{p}) \overset{u}{q}(\vec{x}, t, \vec{p}) + c(\vec{x}, t, \vec{p}) \right) d\partial\Omega dt \tag{5.1a}$$

$$\text{subject to} \quad q = f(\sigma_x^2, \sigma_y^2, \vec{p}), \tag{5.1b}$$

$$\frac{\partial \sigma_x^2}{\partial t} = f(\vec{U}, \vec{x}, \vec{p}), \tag{5.1c}$$

$$\frac{\partial \sigma_y^2}{\partial t} = f(\vec{U}, \vec{x}, \vec{p}), \tag{5.1d}$$

$$\vec{U} = f(\vec{U}_c) + f(\vec{U}_w) + f(\vec{U}_{\text{wave}}) + \vec{U}_{\text{d}}, \tag{5.1e}$$

$$\frac{\partial U_c}{\partial t} = -(U_c \cdot \nabla) U_c + \nu_c \nabla^2 U_c - \nabla w_c + s_c, \tag{5.1f}$$

$$\nabla \cdot U_c = 0, \tag{5.1g}$$

$$\frac{\partial U_w}{\partial t} = -(U_w \cdot \nabla) U_w + \nu_w \nabla^2 U_w - \nabla w_w + s_w, \tag{5.1h}$$

$$\nabla \cdot U_w = 0, \tag{5.1i}$$

$$g(\vec{p}) \leq 0. \tag{5.1j}$$

where the terms in (5.1b) to (5.1d) are defined in (4.85) to (4.86). The function $E(\vec{x}, t, \vec{p})$ where $E : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1 \times 2n_p} \to [0, 1]$, weights the spatio-temporal importance of minimising the uncertainty state tracer $\overset{u}{q}(\vec{x}, t, \vec{p})$. A further term, $c(\vec{x}, t, \vec{p})$, where $c : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1 \times 2n_p} \to \mathbb{R}$ acts as a penalty function for the sensor positions and velocity constraints of (5.1j), defined in Section 5.1.1. Uncertainty drift velocity is $\vec{U}$, as described previously in Section 4.5. Tracer and oil specific uncertainty dynamics are described by (5.1a) to (5.1e), fluid flow constraints in (5.1f) to (5.1i), while sensor constraints are specified by (5.1j). Let $k_\tau \in \mathbb{N}$ be the discrete time index when $t_{k_\tau} = t_\tau$ and $k_\epsilon \in \mathbb{N}$ be the discrete time index when $t_{k_\epsilon} = t_\epsilon$. For a pictorial representation of the optimisation process, refer to Figure 5.1 and Figure 5.2, which display an example of the uncertainty minimisation weighting and distribution respectively.

In the optimisation of (5.1a), the optimisation determines an optimal sensor trajectory $\vec{\mathscr{P}}$ for the time interval $[t_\tau, t_\epsilon]$. If solving for an optimal trajectory is computationally infeasible (this depends on the computing resources available), then a substitution of a one-step-ahead optimisation method is used, as in Zammit Mangion, Anderson, and Kadirkamanathan 2011. In this case, the optimisation determines the optimal sensor positions, where $\vec{\mathscr{P}} = \vec{p}$, for the time

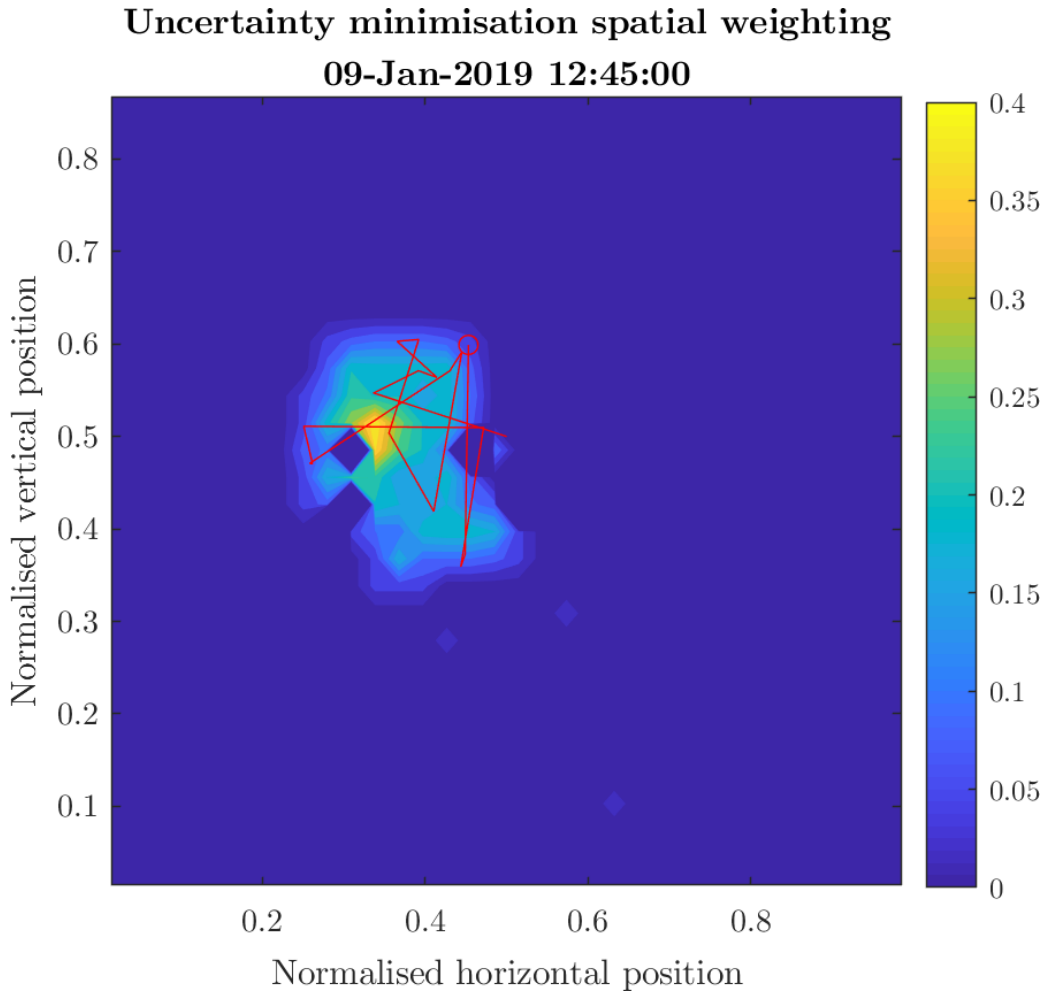


Figure 5.1: This figure displays the spatial weighting of the uncertainty minimisation, $E(\vec{x}, t, \vec{p})$, at a time-step of the monitoring framework application in Chapter 7, Section 7.3. There is high spatial weighting where the spill is expected to be, but also note the non-zero measuring priority in the left figure at (0.62,0.1), (0.58,0.32) and (0.42,0.29) to reduce fluid flow estimation error. Though the spatial weighting at these coordinates is very low, the sensor will begin exploring these areas once the spill has been mapped. The sensor position is displayed as a red circle with a five hour trail. The sensor first flies to the high priority region where the spill is likely to be, mapping upstream to the north, then exploring the down-stream edges of the spill to the west and south.

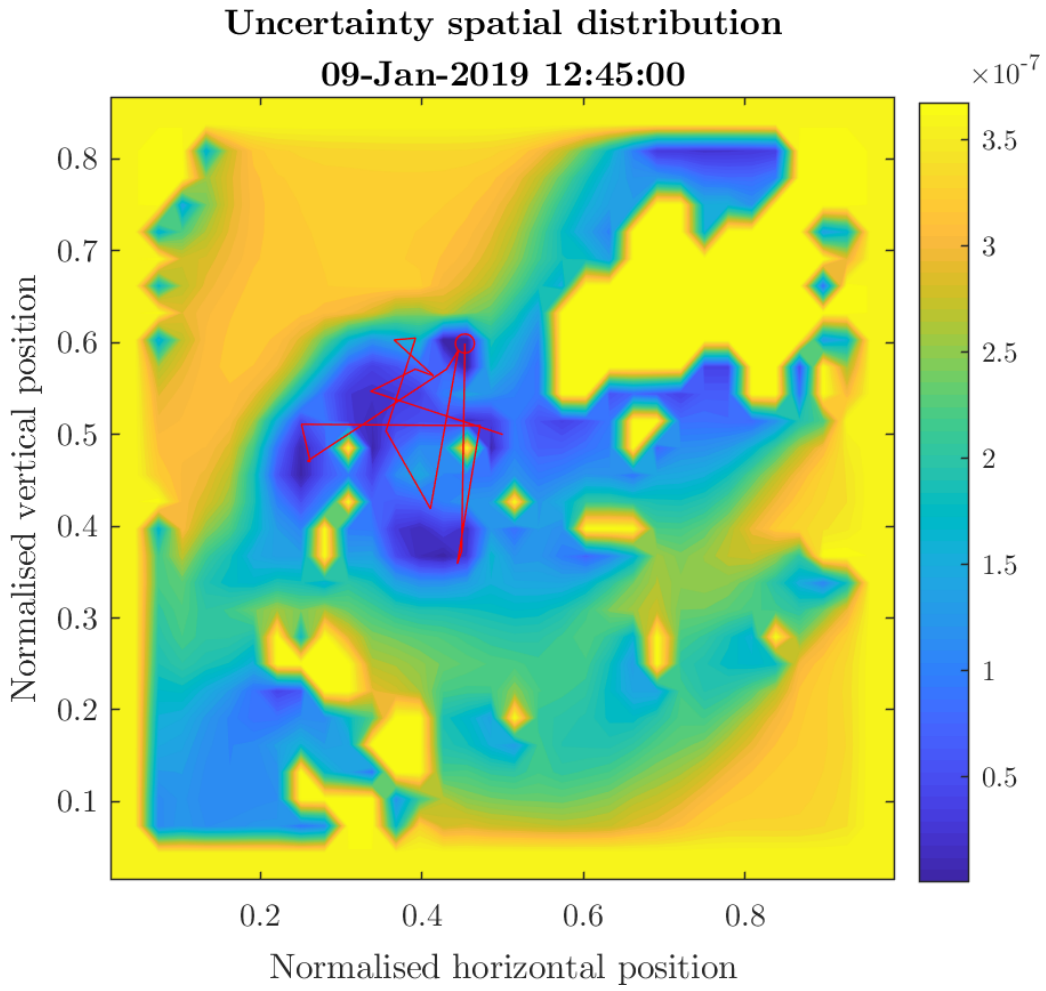


Figure 5.2: This figure displays the spatial distribution of uncertainty, $\overset{u}{q}(\vec{x}, t, \vec{p})$, at a time-step of the monitoring framework application in Chapter 7, Section 7.3. There is low uncertainty where the sensor, displayed as a red circle with a 5 hour trail, has measured and high uncertainty at the domain boundaries spreading across the domain. Areas of land are assigned a high value to represent the complex interaction of flow at a coast-line.

interval $[t_\iota, t_\phi]$. When constrained to one-step-ahead solutions, if a sensor position trajectory is required for the time interval $[t_\tau, t_\epsilon]$ the optimisation is repeated sequentially n_τ times, one-step-ahead each time, up to the time horizon. This forms the trajectory of sensor positions $\vec{\mathscr{P}}$ that are locally optimal in the set of time intervals

$$\{[t_\tau, t_\tau + \delta t], [t_\tau + \delta t, t_\tau + 2\delta t], ..., [t_\tau + (n_o - 1)\delta t, t_\tau + n_o \delta t]\}$$

where $t_\tau + n_o \delta t = t_\epsilon$. This is a trajectory of locally optimal positions, not the optimal trajectory, but is computationally tractable compared to a full adjoint Model Predictive Control (MPC) implementation. In this one-step ahead case, optimal sensor positions are a function of the time-step δt, therefore parallel sequences of adjoint solved optimisations using different time-steps for each position optimisation produce a sensor path up to a common future time, and the lowest total cost function is selected. This parallel method is only worthwhile under memory limitations and if the velocity constraints are affecting the optimal sensor positions. This parallel method is instead of an MPC approach, which is more memory intensive but preferable under constraints. Other cases considered in the following sections include static sensors and full adjoint-MPC.

5.1.1 Uncertainty weighting

The selection of $E(\vec{x}, t, \vec{p})$ that selectively weights particular regions of uncertainty minimisation, allows sensors to prioritise areas and times where the spill probability, or spill probability entropy, or uncertainty in spill location and velocity field are high. The weighting matrix is defined by

$$\begin{aligned} E(\vec{x}, t, \vec{p}) = \frac{1}{k_T} \Bigg(& k_{P_{\hat{\vec{O}}_p}} P_{\hat{\vec{O}}_p}(x_i, y_j) \\ & + k_{S_e} S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j) + k_{P_{\mathrm{KF}}} \mathrm{AMSE}(\hat{\vec{\vec{x}}}) \\ & + k_{S_o} P_{\mathrm{source}}(x_i, y_j) + k_{P_{t-24\mathrm{hr}}} P_{\hat{\vec{O}}_p}(x_i, y_j)_{t-24\mathrm{hr}} \\ & + k_{P_{t-24\mathrm{hr}}} S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j)_{t-24\mathrm{hr}} + k_{\partial\Omega} \Bigg) \end{aligned} \tag{5.2a}$$

where,

$$k_T = k_{P_{\hat{\vec{O}}_p}} + k_{S_e} + k_{P_{\mathrm{KF}}} + k_{S_o} + k_{\partial\Omega} + k_{P_{t-24\mathrm{hr}}}. \tag{5.2b}$$

In (5.2) $S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j)$, where $S_e : \partial\Omega \times \mathbb{R}^{n_s} \times \mathbb{R}_+ \to [0, 1]$, is the min-max normalised (Juszczak, Tax, and Duin 2000) Shannon entropy (Shannon 1948) of $P_{\hat{\vec{O}}_p}(x_i, y_j)$ in the 3-by-3 neighbourhood of the cell at (x_i, y_j). This is a measure of the Shannon self-information in that cell, min-max normalised by the values of S_e across $\partial\Omega$. The entropy term $S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j)$ takes a high value where there is significant spatial variation in spill probability, thus identifying the spill perimeter and internal variation in the spill distribution as measures to determine sensing allocation. The term $E_{\mathrm{KF}}(x_i, y_j, t, \vec{p})$, where $E_{\mathrm{KF}} : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to [0, 1]$, is the min-max normalised augmented state estimation error of the velocity field, arising from the data assimilation method (Chapter 6, Section 6.1.1). Further functions focus on past information, $P_{\mathrm{source}}(x_i, y_j)$, $P_{\hat{\vec{O}}_p}(x_i, y_j)_{t-24\mathrm{hr}}$ and $S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j)_{t-24\mathrm{hr}}$, where $P_{\mathrm{source}} : \partial\Omega \times \mathbb{R}^{n_s} \times \mathbb{R}_+ \to [0, 1]$, $P_{\hat{\vec{O}}_p, t-24\mathrm{hr}} : \partial\Omega \times \mathbb{R}^{n_s} \times \mathbb{R}_+ \to [0, 1]$ and $S_{e,t-24\mathrm{hr}} : \partial\Omega \times \mathbb{R}^{n_s} \times \mathbb{R}_+ \to [0, 1]$. These terms are the probability of presence of the source location of the spill $P_{\mathrm{source}}(x_i, y_j)$, the 24 hour delayed oil probability $P_{\hat{\vec{O}}_p}(x_i, y_j)_{t-24\mathrm{hr}}$ and the 24 hour delayed oil entropy $S_e(P_{\hat{\vec{O}}_p}(x_i, y_j), x_i, y_j)_{t-24\mathrm{hr}}$. Jointly, they provide a quantitative metric with which to determine sensor resource allocation to the likely source location and the oil location 24 hours prior, acquiring the most informative velocity field measurements for that time in the tide cycle, even in asymmetric tide cycles. The total weighting $k_T \in \mathbb{R}_+$ is the sum of the weighting coefficients $k_{P_{\hat{\vec{O}}_p}}, k_{S_e}, k_{P_{\mathrm{KF}}}, k_{S_o}, k_{\partial\Omega}, k_{P_{t-24\mathrm{hr}}} \in \mathbb{R}_+$. Unless otherwise stated, assume a uniform weighting, where $k_{P_{\hat{\vec{O}}_p}}, k_{S_e}, k_{P_{\mathrm{KF}}}, k_{S_o}, k_{P_{t-24\mathrm{hr}}} = 1$, but $k_{\partial\Omega} = 0.001$. This balances sensor effort between spill mapping and exploration, with a balanced temporal focus.

5.1.2 Sensor considerations

Sensor constraints are included in the function of interest J by the penalty function $c(\vec{x}, t, \vec{p})$, defined by

$$\begin{aligned} c(\vec{x}, t, \vec{p}) = V(\vec{p}, \vec{p}_0, v_{\mathrm{sensor}}) + D_m(E(\vec{x}, t, \vec{p}), \vec{p}) \\ + D_e(\partial\Omega, \vec{p}). \end{aligned} \tag{5.3}$$

A velocity penalty term $V(\vec{p}, \vec{p}_{t-\partial t}, v_{\text{sensor}})$, where $V : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}$ maps spatial locations to the euclidean distance to each sensor, though destinations reachable from prior sensor positions $\vec{p}_{t-\partial t}$ within a given time step incur zero penalty. A further term, $D_m(\vec{x}, \vec{p})$, where $D_m : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{n_s} \times \mathbb{R}^{1\times\mathbb{R}^{1\times n_p}} \to \mathbb{R}$, is defined for each sensor position and contains the Euclidean distance to the closest region of interest, where $E(\vec{x}, t, \vec{p}) > 0$. The term $D_e(\partial\Omega, \vec{p})$ where $D_e : \partial\Omega \times \mathbb{R}_+ \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}$ is defined for each sensor position as the Euclidean distance to the closest non-excluded area. The derivatives of V, D_m and D_e with respect to sensor positions are used in the gradient descent solver to guide sensors to feasible, non-empty and permissible regions respectively.

5.2 The adjoint solution method

The optimisation problem (5.1a) is solved through a gradient descent method, using an application of the adjoint method to provide gradient information, in a similar manner to Funke, Farrell, and Piggott 2014. Concisely, the adjoint method constructs the dual form of the constrained optimisation problem, introducing a new adjoint state equation that includes a new, unknown variable, $\vec{\lambda}_k : \mathbb{R}_+ \to \mathbb{R}^{n_u}$. Solving for this dual variable allows construction of the gradient of the original optimisation problem, with respect to the control variables. The adjoint approach first solves the adjoint state equation,

$$\left(\frac{\partial \boldsymbol{F}}{\underset{u}{\partial \vec{\mathscr{X}}}}\right)^* \vec{\mathscr{L}} = \left(\frac{\partial J}{\underset{u}{\partial \vec{\mathscr{X}}}}\right)^*, \tag{5.4}$$

where $\frac{\partial \boldsymbol{F}}{\underset{u}{\partial \vec{\mathscr{X}}}} \in \mathbb{R}^{n_u n_\tau \times n_u n_\tau}$ is the derivative of the state-space system trajectory, described in (4.111). The state-space system describes the set of constraint equations (5.1b) to (5.1j). In (5.4), $(\cdot)^*$ is the conjugate transpose and $\vec{\mathscr{L}} \in \mathbb{R}^{n_u n_\tau}$ is the adjoint solution trajectory formed of n_τ vertically concatenated adjoint state vectors $\vec{\lambda}$. In (5.4), $\underset{u}{\vec{\mathscr{X}}}$ is the state trajectory of the optimisation states and J is the function of interest (5.1a). The cost function derivative is defined $\frac{\partial J}{\underset{u}{\partial \vec{\mathscr{X}}}} \in \mathbb{R}^{1\times n_u n_\tau}$. The next subsection will describe the high level structure of the adjoint equation. The detailed analytical derivatives are described in appendix A. Note that matrix derivatives here and in the appendix use the numerator layout convention, with tensor derivatives as in Khang 2012.

5.2.1 Adjoint equation structure

Introduce the notation $\mathscr{F}(...)^{\underset{\vec{x}_k}{u}} = \frac{\partial \mathscr{F}(...)}{\partial \vec{x}_k}$ where $\mathscr{F}$ is defined in (A.2). Continuing the examples of system trajectories with two time-steps from an initial condition, the matrix description of $\frac{\partial \boldsymbol{F}}{\partial \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}}$ is

$$\frac{\partial \boldsymbol{F}}{\partial \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}} = \begin{bmatrix} 0 & 0 & 0 \\ \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_k, \underset{\underset{\sim}{u}}{\vec{x}}_{k-1}, \vec{p}_{k-1}, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_{k-1}}{u}} & \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_k, \underset{\underset{\sim}{u}}{\vec{x}}_{k-1}, \vec{p}_{k-1}, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_k}{u}} & \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_k, \underset{\underset{\sim}{u}}{\vec{x}}_{k-1}, \vec{p}_{k-1}, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_{k+1}}{u}} \\ \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_{k-1}}{u}} & \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_k}{u}} & \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \vec{\mathscr{P}})^{\underset{\vec{x}_{k+1}}{u}} \end{bmatrix}. \tag{5.5}$$

For brevity, let there be further notation, such that $\mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \mathscr{P})^{\underset{\vec{x}_k}{u}} = \mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k}$. Note that the first subscript $k+1$ refers to the latest time index in $\mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \mathscr{P})$ while the second subscript k is the time index of the derivative variable. Extending this notation, $\mathscr{F}^{\vec{p}}_{k+1,k} = \frac{\partial \mathscr{F}(\underset{\underset{\sim}{u}}{\vec{x}}_{k+1}, \underset{\underset{\sim}{u}}{\vec{x}}_k, \vec{p}_k, \underset{\underset{\sim}{u}}{\vec{\mathscr{X}}}, \mathscr{P})}{\partial \vec{p}_k}$. The adjoint equation (5.4) using (5.5) is described by

$$\begin{bmatrix} 0 & 0 & 0 \\ \mathscr{F}^{\underset{\vec{x}}{u}}_{k,k-1} & \mathscr{F}^{\underset{\vec{x}}{u}}_{k,k} & \mathscr{F}^{\underset{\vec{x}}{u}}_{k,k+1} \\ \mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k-1} & \mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k} & \mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k+1} \end{bmatrix}^* \begin{bmatrix} \vec{\lambda}_{k-1} \\ \vec{\lambda}_k \\ \vec{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial x_{k-1}}^* \\ \frac{\partial J}{\partial x_k}^* \\ \frac{\partial J}{\partial x_{k+1}}^* \end{bmatrix}. \tag{5.6}$$

By inspection of (A.2) and the terms within it, for time-step k to $k+1$, only the derivatives $\mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k}$ and $\mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k+1}$ are non-zero valued. Hence the adjoint equation, with applied conjugate transpose, is again described by,

$$\begin{bmatrix} 0 & \left(\mathscr{F}^{\underset{\vec{x}}{u}}_{k,k-1}\right)^* & 0 \\ 0 & \left(\mathscr{F}^{\underset{\vec{x}}{u}}_{k,k}\right)^* & \left(\mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k}\right)^* \\ 0 & 0 & \left(\mathscr{F}^{\underset{\vec{x}}{u}}_{k+1,k+1}\right)^* \end{bmatrix} \begin{bmatrix} \vec{\lambda}_{k-1} \\ \vec{\lambda}_k \\ \vec{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} \frac{\partial J}{\partial x_{k-1}}^* \\ \frac{\partial J}{\partial x_k}^* \\ \frac{\partial J}{\partial x_{k+1}}^* \end{bmatrix}. \tag{5.7}$$

Note the diagonal structure of the matrix of derivatives in (5.7). Hence an iterative expression can be derived for $\vec{\lambda}$, solving back in time from $\vec{\lambda}_{k_\epsilon}$. The iterative adjoint equation is described

by

$$\left(\mathscr{F}^{\underline{u}\,\vec{\underline{x}}}_{k,k}\right)^{*}\vec{\lambda}_k = \left(\frac{\partial J}{\partial x_k}\right)^{*} - \left(\mathscr{F}^{\underline{u}\,\vec{\underline{x}}}_{k+1,k}\right)^{*}\vec{\lambda}_{k+1}, \tag{5.8}$$

but note this can be rearranged to the following for an inverse based iterative solution:

$$\vec{\lambda}_k = \left(\left(\mathscr{F}^{\underline{u}\,\vec{\underline{x}}}_{k,k}\right)^{*}\right)^{+}\left(\left(\frac{\partial J}{\partial x_k}\right)^{*} - \left(\mathscr{F}^{\underline{u}\,\vec{\underline{x}}}_{k+1,k}\right)^{*}\vec{\lambda}_{k+1}\right), \tag{5.9}$$

where the notation $\cdot^{+}$ is the Moore-Penrose inverse. In practice, it may be preferable to use an efficient linear-solver to determine $\vec{\lambda}_k$ in (5.8). The final time adjoint states $\vec{\lambda}_{k_\epsilon}$ are described by

$$\vec{\lambda}_{k_\epsilon} = \left(\left(\mathscr{F}^{\underline{u}\,\vec{\underline{x}}}_{k_\epsilon,k_\epsilon}\right)^{*}\right)^{+}\left(\frac{\partial J}{\partial x_{k_\epsilon}}\right)^{*}. \tag{5.10}$$

5.2.2 Adjoint gradient calculation

The solving of (5.4) for the trajectory of adjoint states $\vec{\mathscr{L}}$ enables calculation of the cost function gradient by

$$\frac{dJ}{d\vec{\mathscr{P}}} = -\vec{\mathscr{L}}^{*}\frac{\partial \boldsymbol{F}}{\partial \vec{\mathscr{P}}} + \frac{\partial J}{\partial \vec{\mathscr{P}}}, \tag{5.11}$$

where $\frac{dJ}{d\vec{\mathscr{P}}} \in \mathbb{R}^{1\times n_\tau 2n_p}$ is the cost function total derivative with respect to sensor positions, $\frac{\partial \boldsymbol{F}}{\partial \vec{\mathscr{P}}} \in \mathbb{R}^{n_\tau n_u \times n_\tau 2n_p}$ is the partial derivative of the state-space system with respect to sensor positions. The further partial derivative $\frac{\partial J}{\partial \vec{\mathscr{P}}} \in \mathbb{R}^{1\times n_\tau 2n_p}$ describes any direct dependency of the cost function upon sensor positions. In this case, $\frac{\partial J}{\partial \vec{\mathscr{P}}}$ describes the effect of constraints on the cost function.

An expansion of (5.11) to a trajectory containing 5 time-steps is described by,

$$\begin{bmatrix} \frac{\partial J}{\partial \vec{p}_{k-2}} & \frac{\partial J}{\partial \vec{p}_{k-1}} & \frac{\partial J}{\partial \vec{p}_{k}} & \frac{\partial J}{\partial \vec{p}_{k+1}} & \frac{\partial J}{\partial \vec{p}_{k+2}} \end{bmatrix} =$$

$$-\begin{bmatrix} \vec{\lambda}^*_{k-2} & \vec{\lambda}^*_{k-1} & \vec{\lambda}^*_{k} & \vec{\lambda}^*_{k+1} & \vec{\lambda}^*_{k+2} \end{bmatrix} \underbrace{\begin{bmatrix} \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k+2} \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k+2} \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k+2} \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k+2} \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k+2} \end{bmatrix}}_{\frac{\partial \boldsymbol{F}}{\partial \vec{\mathscr{P}}}} + \frac{\partial J}{\partial \vec{\mathscr{P}}}. \tag{5.12}$$

By inspection of (A.2) and the terms within it, for time-step k to $k+1$, only the derivatives $\mathscr{F}^{\vec{p}}_{k+1,k-1}$, $\mathscr{F}^{\vec{p}}_{k+1,k}$ and $\mathscr{F}^{\vec{p}}_{k+1,k+1}$ are non-zero valued. Hence the structure is now restated as,

$$\begin{bmatrix} \frac{\partial J}{\partial \vec{p}_{k-2}} & \frac{\partial J}{\partial \vec{p}_{k-1}} & \frac{\partial J}{\partial \vec{p}_{k}} & \frac{\partial J}{\partial \vec{p}_{k+1}} & \frac{\partial J}{\partial \vec{p}_{k+2}} \end{bmatrix} =$$

$$-\begin{bmatrix} \vec{\lambda}^*_{k-2} & \vec{\lambda}^*_{k-1} & \vec{\lambda}^*_{k} & \vec{\lambda}^*_{k+1} & \vec{\lambda}^*_{k+2} \end{bmatrix} \underbrace{\begin{bmatrix} \mathscr{F}^{\bar{\vec{p}}^*}_{k-2,k-2} & 0 & 0 & 0 & 0 \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k-1,k-1} & 0 & 0 & 0 \\ \mathscr{F}^{\bar{\vec{p}}^*}_{k,k-2} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k,k} & 0 & 0 \\ 0 & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k-1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+1,k+1} & 0 \\ 0 & 0 & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k+1} & \mathscr{F}^{\bar{\vec{p}}^*}_{k+2,k+2} \end{bmatrix}}_{\frac{\partial \boldsymbol{F}}{\partial \vec{\mathscr{P}}}} + \frac{\partial J}{\partial \vec{\mathscr{P}}}. \tag{5.13}$$

5.2.3 One-step-ahead optimisation

In a system where the dynamics of for time-step k is only affected by sensor positions $\vec{p}_k$, then $\frac{\partial F}{\partial \vec{\mathscr{P}}}$ would be non-zero only on the lower diagonal, or $\mathscr{F}^{\vec{p}}_{k+1,k}$. Hence calculation of $\frac{\partial J}{\partial \vec{\mathscr{P}}}$ can be performed iteratively. The iterative adjoint gradient would be described by

$$\frac{dJ}{d\vec{p}_k} = -\vec{\lambda}^*_{k+1} \frac{\partial \mathscr{F}(\overset{u}{\vec{x}}_{k+1}, \overset{u}{\vec{x}}_k, \vec{p}_k, \overset{u}{\vec{\mathscr{X}}}, \mathscr{P})}{\partial \vec{p}_k} + \frac{\partial J}{\partial \vec{p}_k}. \tag{5.14}$$

Here, this iterative approach is only usable for a one-step ahead prediction and sensor-placement, which makes the gradient calculation and optimisation efficient. In a one-step-ahead case, $t_\epsilon = t_\tau + \delta t$, $n_\tau = 1$ and the descriptions in (5.7) and (5.13) reduce in dimension accordingly. The determined gradient is then used in a gradient descent approach to optimisation for the sensor positions one step ahead.

5.2.4 Fixed sensor optimisation

When solving for fixed sensor positions $\vec{p}_{k_\tau}$ that are optimal up to t_ϵ, then only $\mathscr{F}^{\vec{p}}_{k,k_\tau}$ is non-zero and $\vec{p}_k = \vec{p}_{k_\tau}$ for all k. Hence the matrix $\frac{\partial \boldsymbol{F}}{\partial \vec{\mathscr{P}}}$ is non-zero only in the left-most block column in (5.5). It is therefore possible to describe the cost function gradient with respect to the fixed sensor position by

$$\frac{dJ}{d\vec{p}_{k_\tau}} = \sum_{k=k_\tau}^{k=k_\epsilon-1} \left[-\vec{\lambda}^*_{k+1} \frac{\partial \mathscr{F}(\overset{u}{\underline{\vec{x}}}_{k+1}, \overset{u}{\underline{\vec{x}}}_k, \vec{p}_{k_\tau}, \overset{u}{\underline{\vec{\mathscr{X}}}}, \mathscr{P})}{\partial \vec{p}_{k_\tau}} \right] + \frac{\partial J}{\partial \vec{p}_{k_\tau}}. \tag{5.15}$$

This can be determined iteratively, while stepping back to solve for $\vec{\lambda}_k$.

5.2.5 Adjoint MPC

Alternatively, if the memory and computation resources available, the full matrices can be constructed in (5.13) after stepping back in-time and determining $\vec{\mathscr{L}}$, and the gradient descent optimisation will converge to an optimal trajectory of sensor positions. This is an extension of the implementation of adjoint-MPC in Vali et al. 2019 to systems with dependency at time-step k on variables outside of the time-step k.

5.3 Gradient descent optimisation

The initial estimate of sensor placement is at the local maxima of the uncertainty, at each time-step, without sensors. The maxima are ordered from the highest valued maxima to the smallest. Sensors are assigned to each maxima in the fastest feasible configuration. The uncertainty map

without sensors is described by

$$J_t^{\text{empty}} = E(\vec{x}, t)q(\vec{x}, t), \tag{5.16}$$

where the omission of an integral over $\partial\Omega$ or t results in $J_t^{\text{empty}} : \partial\Omega \times \mathbb{R}_+ \to [0, 1]$.
The local maxima are found through a search of the spatial domain for values higher than their immediate neighbours, in the discretised mesh of $n_x n_y$ cells. The initial sensor positions at k are defined by

$$\vec{p}_k = f_p \left(\frac{\int_{t_k}^{t_{k+1}} J_t^{\text{empty}}}{t_{k+1} - t_k}, n_p \right), \tag{5.17}$$

where the function f_p finds the highest n_p number of peaks (one for each sensor) in time-averaged weighted uncertainty and returns their coordinates. In practice, J_t^{empty} is determined for $[t_{k_\tau}, t_{k_\tau+1}]$ and initial sensor positions are determined for $\vec{p}_{k_\tau+1}$ and uncertainty removed accordingly, then J_t^{empty} is determined for $[t_{k_\tau} + 1, t_{k_\tau+2}]$ and initial sensor positions are determined for $\vec{p}_{k_\tau+2}$ and so on, up to $\vec{p}_{k_\epsilon}$. Sensor positions descend the gradient each iteration of the solver, described by

$$d\vec{\mathscr{P}}_n = \gamma_n \frac{dJ}{d\vec{\mathscr{P}}_n}, \tag{5.18}$$

where $n \in \mathbb{N}$ is the iteration index and the step size $\gamma_n : \mathbb{N} \to \mathbb{R}$ is found with a backtracking determined line-search using the Armijo-Goldstein condition (Armijo 1966; Goldstein 1965; Coope and Price 1995). Alternatively, a suitable constant can be used, e.g $\gamma = 0.001$ for all n. Gradient descent continues until $\left\| \frac{dJ}{d\vec{\mathscr{P}}_n} \right\|_2 < \zeta_g$ where $\zeta_g = n_\tau 10^{-3}$ is a threshold value, or descent continues to a maximum number of iterations. The n_τ term in the threshold value scales the threshold value with the number of time-steps in the optimisation, to ensure scalability with longer time-horizons. In practice, given a simple flow and bathymetry, the initial locations identified in (5.17) are suitably close to the optimal that a single gradient step occurs then the stop condition is fulfilled.

Chapter 6

Data assimilation (Stages 3 & 4)

This chapter describes the data assimilation portion of the framework. This work adopts an approach with Kalman estimators and Dynamic-Mode-Decomposition (DMD) (Schmid 2010) similar to Nonomura, Shibata, and Takaki 2019, employing a Rauch-Tung-Striebel Smoother (akin to a dual pass Kalman filter and Smoother) and various extensions of DMD with a historic state-trajectory and iterative improvement with a moving horizon. DMD extracts coherent spatial-temporal patterns, as a mode matrix and a dynamics matrix, from a state-trajectory and forms a low-order model to approximate the dynamics of SCEM, repeated upon a moving time-window of the SCEM state trajectory.

6.1 Forming the reduced order model (Stage 3)

This section describes the identification of the reduced order model used for data assimilation. The solution, or state-trajectory, of the non-linear state equations within the high order model can be represented by an LTI system of infinite state dimension, as described by Koopman Operator Theory (Koopman 1931; Schmid 2010; Tu et al. 2014; Brunton et al. 2015). In practice, the Koopman Operator is often approximately low-rank. DMD is a computationally efficient procedure for identifying the dominant eigenfunctions and eigenvalues of the Koopman Operator from simulation data. These can then be used to form a reduced-order model, enabling efficient estimation of the full-order system states, domain-wide, from sparse localised measurements. The reduced-order, linear, and time-invariant model enables efficient hind-casting and prediction of fluid states. The approach here utilises a Centred (Hirsh et al. 2020) Hankel

(Filho and Lopes 2019) weighted Dynamic Mode Decomposition of the high order model simulation. This decomposition utilises an augmented time-window of the SCEM state trajectory, described next.

DMD trajectory construction

The DMD initial data set is a time subset, between sensor activation time $t_s \in \mathbb{R}_+$ and the latest sensing time $t_c \in \mathbb{R}_+$ of the state trajectory and is defined by

$$\mathscr{D} = \mathscr{X}_{t_s \to t_c}, \tag{6.1}$$

where t_s and t_c are defined such that $t_c - t_s$ selects the period sensors have been active, or t_s is set to be 24 hours prior to t_c if sensors have been active less than 12 hours and hence would include purely external data forced state snapshots in the trajectory, from before sensors arrived. This is to ensure at least one complete tide cycle has been observed by sensors, or in past data, for a reduced order model to utilise. To enforce 24 hour cycles in flow from a data set less than 24 hours long, e.g if sensors have been active 15 hours, the DMD trajectory is appended with the beginning states, moved 24 hours ahead in time. Let $t_e \in \mathbb{R}_+$ where $t_e = t_s + 24_{\text{hrs}}$, then the trajectory is appended as described by,

$$\mathscr{D}_c = [\mathscr{D}, \vec{x}_{t_s}]. \tag{6.2}$$

Similarly, a corresponding time trajectory is constructed,

$$\mathscr{T}_c = [\mathscr{T}_{t_s \to t_c}, t_e]\,. \tag{6.3}$$

The data is then centred through linear regression.

Alternatively, if the data set is longer than 24 hours, the linear regression is performed prior to enforcing cyclic data, and on a subset containing the last multiple of 24 hours of data, (24, 48, 72 hours etc). The appended point is then placed at the next complete day after t_c, at 48 hours, 72 hours etc after t_s, at time t_e. The appended point has values extrapolated from the linear regression. This ensures the low order model captures slow growth or decay from spring or neap tides, without confusion from mixed semi-diurnal (irregular) tides.

The trajectory of each state is centred around a first order linear regression of that state against time, solving (6.4a) through ordinary least squares. This is an extension of Hirsh et al. 2020 where states are centred around their mean value. A state x_n, with the DMD trajectory $\hat{\mathscr{D}}_{c_n}$, has regression coefficients α_n and β_n that solve the minimisation,

$$\underset{\alpha_n,\,\beta_n}{\text{minimize}} \quad (\mathscr{D}_{c_n} - \alpha_n - \beta_n \mathscr{T}_c)^2. \tag{6.4a}$$

The regression coefficient matrices are constructed,

$$\vec{\alpha} = [\alpha_0, \alpha_1, ..., \alpha_{n_s}]^T, \tag{6.5a}$$

$$\vec{\beta} = [\beta_0, \beta_1, ..., \beta_{n_s}]^T. \tag{6.5b}$$

The centred trajectory is then defined by,

$$\bar{\mathscr{D}} = \mathscr{D}_c - \vec{\alpha} - \vec{\beta} \circ \mathscr{T}_{k|k}, \tag{6.6}$$

where $\circ$ is the Hadamard product. The centred trajectory $\bar{\mathscr{D}}$ is formed of centred state vectors $\bar{\vec{x}}_t : \mathbb{R}_+ \to \mathbb{R}^{n_T}$. Similarly, the sensor measurements of velocity $\vec{y}(\vec{p}, t) : \mathbb{R}^{1\times 2n_p} \times \mathbb{R}_+ \to \mathbb{R}^{2n_p}$ are also centred to $\bar{\vec{y}}(\vec{p}, t)$,

$$\bar{\vec{y}}(\vec{p}, t) = \vec{y}(\vec{p}, t) - \vec{\alpha}_y - \vec{\beta}_y \circ t, \tag{6.7}$$

where $\vec{\alpha}_y$ and $\vec{\beta}_y$ are the subsets of $\vec{\alpha}$ and $\vec{\beta}$ that correspond to the sensor locations. External data is $\vec{k}_t \in \mathbb{R}^{n_k}$ are the centred external data states of time varying size $n_k : \mathbb{R}_+ \to \mathbb{N}$. The external data is centred in the same manner,

$$\bar{\vec{k}}_t = \vec{k}_t - \vec{\alpha}_k - \vec{\beta}_k \circ t, \tag{6.8}$$

where $\vec{\alpha}_k$ and $\vec{\beta}_k$ are the subsets of $\vec{\alpha}$ and $\vec{\beta}$ that correspond to the external data location that do not overlap with sensor locations.

The DMD trajectory $\bar{\mathscr{D}}$ is now a set of zero mean states from t_s to t_e. However, note that the data is non-uniform in time because SCEM does not utilise a fixed time-step, and there is

a gap in the state trajectory between t_c and t_e. In a similar manner to Leroux and Cordier 2016, although utilising a least squares two-term Fourier fit model instead of an Expectation-Maximisation KF estimated Gaussian model, the missing centred state values between t_c and t_e are estimated. In short, this is a Fourier fit to the trajectory of each single state, extrapolated and sampled to provide values for the state between t_c and t_e. The Fourier fit is appropriate given the centred data and assumption of periodic flow. Traditional DMD requires regularly sampled data, so a new time trajectory is constructed with a fixed time-step $\Delta t \in \mathbb{R}_+$,

$$\mathscr{T}_r = [t_s, t_s + \Delta t, t_s + 2\Delta t, ..., t_e]. \tag{6.9}$$

A corresponding regular centred state trajectory $\bar{\mathscr{D}}_r$ is also constructed. Linear interpolation of the centred state trajectory is used from t_s to t_c, while the Fourier fit is used to generate data points between t_c and t_e. The regular data matrix is defined by,

$$\bar{\mathscr{D}}_r = [\bar{\vec{x}}_{t_s}, \bar{\vec{x}}_{t_s+\Delta t}, \bar{\vec{x}}_{t_s+2\Delta t}, ..., \bar{\vec{x}}_{t_e}]. \tag{6.10}$$

Application of Hankel data matrices to the data set reduces the number of samples available (see Section 6.1). Hence the data set is extended by repetition, $n_r \in \mathbb{N}$ times, where $n_r = 3$ is chosen for Section 7.2. A 3 day long data set will contain a complete tide-cycle, even if data at the beginning and end is lost. For $n \in [\![1, n_r]\!]$, the extended state and time trajectories are defined by,

$$\bar{\mathscr{D}}_e = [\underbrace{\bar{\mathscr{D}}_r, \bar{\mathscr{D}}_r, ..., \bar{\mathscr{D}}_r}_{n_r \text{ times}}], \tag{6.11a}$$

$$\mathscr{T}_e = [\mathscr{T}_r, (n-1)t_e + \mathscr{T}_r - t_s, ..., (n_r-1)t_e + \mathscr{T}_r - t_s]. \tag{6.11b}$$

Hankel DMD

The DMD will fail to extract the dominant mode present in an oscillatory signal if the set of data containing those signals has cardinality less than the number of these modes (Filho and Lopes 2019). Consider a data set described by,

$$\theta(t) = cos(\omega_0 t) = \frac{1}{2}\left(exp(j\omega_0 t) + exp(-j\omega_0 t)\right), \tag{6.12}$$

where j is the imaginary $\sqrt{-1}$. The DMD of the data set produces a first order system with one mode, while the true system contains two modes (Filho and Lopes 2019). Due to the oscillatory nature of tidal and wind flow, it is possible that every state in $\bar{\mathscr{D}}_r$ is an oscillatory signal with more than one mode and hence the cardinality of the data set must be extended. To ensure the DMD accurately captures the oscillatory behaviour, and allow use of past data and measurements to determine the phase of the oscillatory flow, a time-lagged data set in the form of a Hankel matrix is constructed. The Hankel matrix is defined by,

$$\bar{\mathscr{H}}_r = \begin{bmatrix} \{\bar{D}_e | t_i \leq t \leq t_e - n_t k_d \Delta t\} \\ \{\bar{D}_e | t_i + k_d \Delta t < t \leq t_e - (n_t - (n-1)) k_d \Delta t\} \\ \vdots \\ \{\bar{D}_e | t_i + (n-1) k_d \Delta t < t \leq t_e - (n_t - (n-1)) k_d \Delta t\} \\ \vdots \\ \{\bar{D}_e | t_i + (n-1) k_d \Delta t < t \leq t_e - k_d \Delta t\} \\ \{\bar{D}_e | t_i + (n-1)) k_d \Delta t < t \leq t_e\} \end{bmatrix} \tag{6.13}$$

where $k_d \in \mathbb{N}$ defines the time lag multiple of Δt for each row and $n \in [\![1, n_t]\!]$ is the lag index beginning at 1 for the first row and stepping to $n_t \in \mathbb{N}$ for the last row. The Hankel matrix $\bar{\mathscr{H}}_r \in \mathbb{R}^{h_r \times h_c}$ has $h_r = n_T n_t$ rows and $h_c = \frac{(n_r t_e - t_s) - t_s}{\Delta t} - n_t$ columns. Recall that n_T is the number of states in the high-order system. For an example data set, $\bar{\mathscr{D}}_r = [\bar{\vec{x}}_{t_1}, \bar{\vec{x}}_{t_2}, \bar{\vec{x}}_{t_3}, \bar{\vec{x}}_{t_4}, \bar{\vec{x}}_{t_5}]$, with $k_d = 1$ and $n_t = 3$, a corresponding Hankel matrix $\bar{\mathscr{H}}_r$ is defined by,

$$\bar{\mathscr{H}}_r = \begin{bmatrix} \bar{\vec{x}}_{t_1} & \bar{\vec{x}}_{t_2} & \bar{\vec{x}}_{t_3} \\ \bar{\vec{x}}_{t_2} & \bar{\vec{x}}_{t_3} & \bar{\vec{x}}_{t_4} \\ \bar{\vec{x}}_{t_3} & \bar{\vec{x}}_{t_4} & \bar{\vec{x}}_{t_5} \end{bmatrix}. \tag{6.14}$$

Note the reduction in samples of the data set from 5 to 3, but increase in cardinality. The selection of values for k and n_t allow for inclusion of more delayed data, without extending cardinality of the data set so much that the singular value decomposition becomes computationally prohibitive in the real-time system.

Recalling the notation in Chapter 1, describe two data sets, $\bar{\mathscr{H}}'$ and $\bar{\mathscr{H}}$, as follows

$$\bar{\mathscr{H}}' = \boldsymbol{A} \bar{\mathscr{H}}, \tag{6.15}$$

and

$$\bar{\mathscr{H}}' = \bar{\mathscr{H}}_r(\llbracket 1, h_r \rrbracket, \llbracket 2, h_c \rrbracket), \tag{6.16a}$$

$$\bar{\mathscr{H}} = \bar{\mathscr{H}}_r(\llbracket 1, h_r \rrbracket, \llbracket 1, h_c - 1 \rrbracket), \tag{6.16b}$$

where $\bar{\mathscr{H}}', \bar{\mathscr{H}} \in \mathbb{R}^{h_r \times h_c - 1}$ are the Hankel matrices formed from the centred state trajectories and $\boldsymbol{A} \in \mathbb{R}^{h_r \times h_r}$ is the state transition matrix. The notation $\bar{\cdot}$ represents centred values, while $\hat{\cdot}$ are estimates, unless stated otherwise. The number of rows and columns in the Hankel matrices are $h_r \in \mathbb{N}$ and $h_c \in \mathbb{N}$ respectively. In practice, wind and ocean velocities are assimilated separately, and trajectories are formed using sequences of the $(n_x + 1)n_y + n_x(n_y + 1)$ wind or ocean velocity component states. For a velocity represented by 2 states, the number of data rows h_r is approximately 5×10^5 in the simulations of Section 7.2. Hence the dimensions $\boldsymbol{A}$ prohibits a least-squares solution for $\boldsymbol{A}$ in (6.15). Instead, the DMD computes the subset of eigenvectors and eigenvalues of the $\boldsymbol{A}$ that minimises $\left\| \bar{\mathscr{H}}' - \boldsymbol{A}\bar{\mathscr{H}} \right\|_F$, without explicitly forming or storing $\boldsymbol{A}$. In comparison, just 5 low-order states are used in Section 7.2.

Reduced order model definition

In traditional DMD, the first step is to identify the Proper Orthogonal (POD) modes of the data-set using an economy or truncated Singular Value Decomposition (SVD). The SVD constructs a truncated rank $n_z \in \mathbb{N}$ approximant of $\bar{\mathscr{H}}$, $\bar{\mathscr{H}} \approx \boldsymbol{U}\boldsymbol{S}\boldsymbol{V}^*$, with the left unitary matrix $\boldsymbol{U} \in \mathbb{R}^{h_r \times n_z}$, the singular value matrix $\boldsymbol{S} \in \mathbb{R}^{n_z \times n_z}$ and the right unitary matrix $\boldsymbol{V}^* \in \mathbb{R}^{n_z \times h_c - 1}$. However, the POD modes are determined by the Hankel state trajectory, which may contain inaccurate states from poor prior predictions or external data. This work utilises DMD with Bi-Iterative Regularised SVD (BIRSVD) (Dasa and Neumaiera 2011), where a weighting matrix $\boldsymbol{W} \in [0, 1]^{h_r \times h_c}$ applies an individual weighting to each state in the Hankel matrix. The state trajectory is weighted in a manner that encourages the modes to describe well estimated parts of the state-trajectory, while ignoring states with a high degree of uncertainty that may result in inaccurate mode shapes.

The BIRSVD also results in a truncated rank approximant of $\bar{\mathscr{H}}$, with a numerically different but structurally identical decomposition. Continuing with the DMD (Schmid 2010) to identify the reduced order system, an identified reduced order system dynamics matrix $\boldsymbol{A}_r \in \mathbb{R}^{n_z \times n_z}$

is constructed by $\boldsymbol{A}_r = \boldsymbol{U}\bar{\mathscr{H}}'\boldsymbol{V}\boldsymbol{S}^{-1}$. The DMD approximates the eigenvectors (DMD modes) of the high order dynamics matrix $\boldsymbol{A}$, from the eigenvectors of $\boldsymbol{A}_r$, since the former represent basis functions from which to reconstruct the air/ocean velocity fields. Compute the eigen decomposition $\boldsymbol{A}_r\boldsymbol{\Upsilon} = \boldsymbol{\Lambda}\boldsymbol{\Upsilon}$, where $\boldsymbol{\Upsilon} \in \mathbb{C}^{n_z \times n_z}$ is formed by column stacking the eigenvectors of $\boldsymbol{A}_r$ and $\boldsymbol{\Lambda} \in \mathbb{C}^{n_z \times n_z}$ is a diagonal matrix of eigenvalues. If a strong tidal component is expected in the flow dynamics, the eigenvalues contained in $\boldsymbol{\Lambda}$ are modified such that their oscillation period is a divisor of the 24 hour tide cycle. For example, consider the ith eigenvalue $\boldsymbol{\Lambda}_i \in \mathbb{C}$ and the jth eigenvalue $\boldsymbol{\Lambda}_j \in \mathbb{C}$ which form a conjugate pair of discrete domain oscillatory complex valued eigenvalues, where $i, j \in [\![1, n_z]\!]$. The modification will be described for the ith eigenvalue, but a similar process is used for the jth and all others. Let the eigenvalue $\boldsymbol{\Lambda}_i$ be mapped to the continuous domain by

$$\boldsymbol{\Lambda}_i^t = \frac{\ln(\boldsymbol{\Lambda}_i)}{\Delta t}, \tag{6.17}$$

where $\ln(\cdot)$ is the natural logarithm and $\boldsymbol{\Lambda}_i^t \in \mathbb{C}$ is the continuous domain eigenvalue. The desired oscillation period is $\mathrm{T}_d \in \mathbb{R}_+$ and the eigenvalue oscillation period $\mathrm{T}_i \in \mathbb{R}_+$ is described by

$$\mathrm{T}_i = \frac{2\pi}{|\boldsymbol{\Lambda}_i^t|}, \tag{6.18}$$

where $|\cdot|$ is the absolute value, or magnitude, of the complex value. The modified continuous domain eigenvalue $\tilde{\boldsymbol{\Lambda}}_i^t \in \mathbb{C}$ is then,

$$\tilde{\boldsymbol{\Lambda}}_i^t = \frac{\mathrm{T}_d}{\mathrm{T}_i}\boldsymbol{\Lambda}_i^t. \tag{6.19}$$

Finally, the discrete domain eigenvalue is revalued by,

$$\boldsymbol{\Lambda}_i = e^{\boldsymbol{\Lambda}_i^t}, \tag{6.20}$$

and substituted back into $\boldsymbol{\Lambda}$ in the appropriate location. In practice, eigenvalues with a period of oscillation within 1 hour of a 6, 12 or 24 hour period, are modified to be eigenvalues with 6, 12 and 24 hour periods of oscillation.

The reduced order system has the mode matrix $\boldsymbol{\Phi} \in \mathbb{C}^{h_r \times n_z}$, where each column is a mode

mapping. The orthogonal modes from the BIRSVD are transformed to DMD modes by,

$$\boldsymbol{\Phi} = \boldsymbol{U}\boldsymbol{\Upsilon}. \tag{6.21}$$

Note that $\boldsymbol{\Phi}$, obtained from a truncated decomposition, is a linear transformation matrix that maps between the complex states of the reduced order system and the real states of the high order system. Without restricting the set of modes, the DMD mode amplitudes are the entries of a matrix $\boldsymbol{z}_r \in \mathbb{C}^{n_z \times h_c}$ of complex random variables, such that

$$\bar{\mathscr{H}}_r \approx \boldsymbol{\Phi}\boldsymbol{z}_r, \tag{6.22}$$

and the full Hankel data matrix is constructed when mapping into the high order state space. The matrix of DMD mode amplitudes $\boldsymbol{z}_r$ is a column stacked trajectory, $\mathscr{Z} \in \mathbb{C}^{n_z \times h_c}$, of mode amplitude vectors $\vec{z}_t : \mathbb{R}_+ \to \mathbb{C}^{n_z}$. For estimation of the current-time flow fields, only a vector description of current-time mode amplitudes $\vec{z}_t$ is required. These mode amplitudes map to a single column of the Hankel data matrix $\bar{\mathscr{S}} \in \mathbb{R}^{h_r}$, described by

$$\bar{\mathscr{S}} = \begin{bmatrix} \bar{\vec{x}}_{t-n_d k_d \delta t} \\ \vdots \\ \bar{\vec{x}}_{t-(n_t-(n-1))k_d \delta t} \\ \vdots \\ \bar{\vec{x}}_t \end{bmatrix} \approx \boldsymbol{\Phi}\vec{z}_t, \tag{6.23}$$

which, as a trajectory of $\bar{\vec{x}}_t$, allows use of historic data. The phase and amplitude accuracy of the estimated DMD modes is improved by the inclusion of prior estimated states and measurements, which provide information to determine the tidal phase. It is necessary to describe a trajectory of all available centred sensor data $\bar{\mathscr{Y}} \in \mathbb{R}^{2n_m n_p}$ and all available external data $\bar{\mathscr{K}} \in \mathbb{R}^{n_k}$, for

the times in $\bar{\mathscr{S}}$,

$$\bar{\mathscr{Y}} = \begin{bmatrix} \bar{\vec{y}}_{t-n_d k_d \delta t} \\ \vdots \\ \bar{\vec{y}}_{t-(n_t-(n-1))k_d \delta t} \\ \vdots \\ \bar{\vec{y}}_t \end{bmatrix}, \bar{\mathscr{K}} = \begin{bmatrix} \bar{\vec{k}}_{t-n_d k_d \delta t} \\ \vdots \\ \bar{\vec{k}}_{t-(n_t-(n-1))k_d \delta t} \\ \vdots \\ \bar{\vec{k}}_t \end{bmatrix}, \tag{6.24}$$

where $\bar{\vec{y}} : \mathbb{R}_+ \to \mathbb{R}^{2n_p}$ are centred sensor measurements, $\bar{\vec{k}} : \mathbb{R}_+ \to \mathbb{R}^{n_{k_t}}$ is the centred external data of dimension $n_{k_t} : \mathbb{R}_+ \to \mathbb{N}$, $n_m \in \mathbb{N}$ is the number of times the sensors have measured and $n_k \in \mathbb{N}$ is the amount of external data states available. Where the available sensor measurements and external data overlap, only the sensor measurements are kept and the corresponding external data is removed from $\bar{\mathscr{K}}$.

The reduced order system has a state equation of,

$$\vec{z}_{t+\delta t} = \boldsymbol{\Lambda}^{\frac{\delta t}{\Delta t}} \vec{z}_t, \tag{6.25}$$

where it should be stated that $\vec{z}_t = \vec{z}_k$, and $\vec{z}_{t+\delta t} = \vec{z}_{k+1}$. This is just equivalence of continuous time and discrete time notation. There is an output equation of

$$\begin{bmatrix} \bar{\mathscr{Y}} \\ \bar{\mathscr{K}} \end{bmatrix} = \boldsymbol{C} \vec{z}_t, \tag{6.26}$$

where $\boldsymbol{C} : \mathbb{R}^{1 \times 2n_p} \to \mathbb{C}^{(2n_m n_p + n_k) \times n_z}$ is the output and sensor matrix, further defined in Section 6.1.1. Note that (6.26) produces centred values. The output equation for non-centred values is described by

$$\begin{bmatrix} \mathscr{Y} \\ \mathscr{K} \end{bmatrix} = \boldsymbol{C} \vec{z}_t + \vec{\alpha}_{y,k} + \vec{\beta}_{y,k} \circ t, \tag{6.27}$$

where $\vec{\alpha}_{y,k} \in \mathbb{R}^{2n_m n_p + n_k}$ and $\vec{\beta}_{y,k} \in \mathbb{R}^{2n_m n_p + n_k}$ are the regression coefficients for the locations corresponding to measured and external data in the state vector.

In the simulations of Section 7.2, the reduced order model captures the evolution of the a 2D flow-field in a large coastal area with complex geometry, using 5 states, instead of approximately 20000. This reduced order representation allows an estimation of the low-order states

to reconstruct an entire flow-field and efficient past analysis or future prediction by stepping the reduced order model through time, as will be detailed in following sections. Though analysis is used here, as in fluid assimilation literature, hindcasting is also an appropriate term.

6.1.1 State estimation (Stage 4)

The reduced order model (6.25) is utilised for estimation of mode amplitudes, $\hat{\vec{z}}_t \in \mathbb{C}^{n_z}$, which in turn allow very sparse sensor measurements of flow velocity to be used to estimate the entire flow field. The reduced order model states are determined using an RTS smoother, which consists of a forward pass time-varying Kalman filter, and a backward recursion smoother, using sensor data and previous states as measurements. The majority of error in data assimilation stems from poor reduced order model mode shapes, and the poor mode shapes form from an inaccurate state trajectory that improves as the monitoring framework iterates with new sensor readings. Therefore, as the RTS smoother is solely concerned with the reduced order mode amplitudes, a smoothed estimate of the mode amplitudes only occurs prior to a long-term analysis or prediction to ensure the most accurate initial conditions for state extrapolation. RTS smoother produced states are referred to as state estimates, not smoothed state estimates, as only the forward pass time-varying Kalman filter is active for the vast majority of the framework iterations. The estimated modal amplitudes, the low order states, are then used to reconstruct the states of the high order system.

The Rauch-Tung-Striebel smoother

The RTS smoother sensor matrix in (6.26) makes use of the current and previous time centred sensor data, and centred external data if available, to form first an estimate, then a smoothed estimate of reduced order modal states prior to extrapolation.

Recall that $\mathbf{\Phi}$ is a tall thin matrix, that provides an approximate linear transform from low order states to a column of the hankel data matrix of high order states, as in (6.23). By selecting a subset of rows in $\mathbf{\Phi}$, only a subset of the Hankel data matrix is constructed. For example, if only the last n_s rows of $\mathbf{\Phi}$ are selected, only $\bar{\vec{x}}_t$ would be constructed in (6.23). Let $\vec{\mathscr{I}}_{\bar{\mathscr{X}}} \in \mathbb{N}^{n_s}$

be described by $\vec{\mathscr{I}}_{\bar{\mathscr{X}}} = [\![h_r - n_s, h_r]\!]$, then

$$\bar{\vec{x}}_t \approx \boldsymbol{\Phi}(\!|\vec{\mathscr{I}}_{\bar{\mathscr{X}}}, [\![1, n_z]\!]|\!)\vec{z}_t. \tag{6.28}$$

In (6.28), $\bar{\vec{x}}_t \in \mathbb{R}^{n_s}$ is the centred state vector. For brevity, let $\boldsymbol{\Psi} \in \mathbb{C}^{n_s \times n_z}$ be defined by $\boldsymbol{\Psi} \coloneqq \boldsymbol{\Phi}(\!|\vec{\mathscr{I}}_{\bar{\mathscr{X}}}, [\![1, n_z]\!]|\!)$. If estimated states are utilised instead, then

$$\hat{\bar{\vec{x}}}_t = \boldsymbol{\Psi}\hat{\vec{z}}_t, \tag{6.29}$$

where $\hat{\bar{\vec{x}}}_t \in \mathbb{R}^{n_s}$ and $\hat{\vec{z}}_t \in \mathbb{R}^{n_z}$ are the estimated centred full order and reduced order state vectors respectively.

Define $\vec{\mathscr{I}}_{\bar{\mathscr{Y}}} \in \mathbb{N}^{n_m}$ as the n_m indices of $\bar{\mathscr{S}}$ that correspond to the spatio-temporal locations of the measurements in $\bar{\mathscr{Y}}$. Further define $\vec{\mathscr{I}}_{\bar{\mathscr{K}}} \in \mathbb{N}^{n_k}$ as the n_k indexes of $\bar{\mathscr{S}}$ that correspond to the spatio-temporal locations of the external data in $\bar{\mathscr{K}}$. Finally, define $\vec{\mathscr{I}}_{\bar{\mathscr{A}}} \in \mathbb{N}^{n_m+n_k}$ as the combined indexes, $\vec{\mathscr{I}}_{\bar{\mathscr{A}}} = \vec{\mathscr{I}}_{\bar{\mathscr{Y}}} \cup \vec{\mathscr{I}}_{\bar{\mathscr{K}}}$. Then the reduced order system sensor matrix, $\boldsymbol{C} : \mathbb{N} \times \mathbb{R}^{2n_p} \to \mathbb{C}^{n_m+n_k \times n_z}$, is defined by

$$\boldsymbol{C} \coloneqq \boldsymbol{\Phi}(\!|\vec{\mathscr{I}}_{\bar{\mathscr{A}}}, [\![1, n_z]\!]|\!). \tag{6.30}$$

The sensor matrix $\boldsymbol{C}$ is a function of more than just the discrete time-step k and sensor positions $\vec{p}$, but the further dependencies upon the state-trajectory, external data and past measurements are ignored for brevity and are contained in the time-step k regardless. The further sensor matrix $\boldsymbol{C}_{\mathrm{KF}} \in \mathbb{C}^{2n_p \times n_z}$ contains the rows of $\boldsymbol{\Phi}$ corresponding to the current-time sensor measurements only, such that $\bar{y}_t = \boldsymbol{C}_{\mathrm{KF}}\vec{z}_t$. This is used in the adjoint derivatives in Appendix A. This substantially cuts down on the dimensions of matrices involved and thus speeds up computation time. Furthermore, the adjoint method derivatives, and sensor-placement optimisation in general, is concerned with sensor placement; The only changeable sub-matrix in $\boldsymbol{C}$ is $\boldsymbol{C}_{\mathrm{KF}}$ at a given time-step. This changes the dimensions of the Kalman filter variables when the optimisation model steps ahead to a future time-horizon.

There are two time-varying Kalman filters, both utilising the reduced order model. The full Kalman filter is part of the RTS smoother, described in this section, and this assimilates trajectories of sensor measurements and external data to provide an estimate of the reduced

order states. Then a reduced Kalman filter, which is identical, but only utilises the current-time sensor measurements and operates solely within the uncertainty system used in the optimisation (5.1a). In the uncertainty system, the estimation of the reduced order model states is not required: the sensor pathing is in the future and sensor measurements are hypothetical. Hence, this reduced Kalman filter only estimates the effect of future sensor placement on the data-assimilation error $E_{\mathrm{KF}_x}(\vec{p})$ and $E_{\mathrm{KF}_y}(\vec{p})$ in (4.86), without the increased complexity of including external or historic data. The differences in the reduced Kalman filter are the substitution of $\boldsymbol{C}_{\mathrm{KF}}$ for $\boldsymbol{C}$, the measurement covariance matrix $\boldsymbol{R}_{\mathrm{KF}} \in \mathbb{R}^{2n_p \times 2n_p}$ is for the current-time sensor measurements alone, and finally the absence of the reduced order model state estimate update equation (6.31).

The RTS smoother is split into two parts: A time-varying Kalman filter in the forward pass, where the prior and posterior state estimates and covariances are saved for use in the backwards pass smoother. Let $\hat{\vec{z}}_{t|k-1} \in \mathbb{C}^{n_z}$ and $\hat{\vec{z}}_{t|k} \in \mathbb{C}^{n_z}$ be the prior and posterior state estimates, and then $\boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1})_{t|k-1} : \mathbb{R}_+ \times \mathbb{R}^{2n_p} \to \mathbb{C}^{n_z \times n_z}$ and $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k} : \mathbb{R}_+ \times \mathbb{R}^{2n_p} \to \mathbb{C}^{n_z \times n_z}$ are the corresponding prior and posterior estimated error covariance matrices. Let $\boldsymbol{R}_e \in \mathbb{R}^{n_z \times n_z}$ and $\boldsymbol{R}_v \in \mathbb{R}^{n_m+n_k \times n_m+n_k}$ be the state and measurement covariance matrices respectively. The measurement update of the forward time-varying Kalman filter is described by

$$\hat{\vec{z}}_t = \hat{\vec{z}}_{t|k} = \hat{\vec{z}}_{t|k-1} + \boldsymbol{L}_k \left(\begin{bmatrix} \bar{\mathscr{Y}} \\ \bar{\mathscr{K}} \end{bmatrix} - \boldsymbol{C} \hat{\vec{z}}_{t|k-1} \right), \tag{6.31}$$

where $\boldsymbol{L}_k : \mathbb{N} \times \mathbb{R}^{2n_p} \to \mathbb{C}^{n_z \times n_m+n_k}$ is the Kalman gain and is described by

$$\boldsymbol{L}_k = (\boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1})_{t|k-1}\boldsymbol{C})(\boldsymbol{C}\boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1})_{t|k-1}\boldsymbol{C}^* + \boldsymbol{R}_v)^{-1}. \tag{6.32}$$

The prior estimated error covariance matrix is updated to the posterior matrix by

$$\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k} = (\boldsymbol{I} - \boldsymbol{L}_k\boldsymbol{C})\boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1})_{t|k-1}. \tag{6.33}$$

A new set of priors for the next measurement time is determined by

$$\hat{\vec{z}}_{t+\delta t|k} = \boldsymbol{\Lambda}^{\frac{\delta t}{\Delta t}} \hat{\vec{z}}_{t|k}, \tag{6.34}$$

and

$$\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t+\delta t|k} = \boldsymbol{\Lambda}^{\frac{\delta t}{\Delta t}} \boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k} (\boldsymbol{\Lambda}^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_e. \tag{6.35}$$

The covariance of the reduced order model states $\boldsymbol{R}_e$ is estimated through application of the sample covariance formula (Montgomery and Runger 1994) to the trajectory $\mathscr{Z}$, for use in the RTS smoother. Also required for the RTS smoother, covariance values for sensor readings, past sensor readings and external data should be provided for use in $\boldsymbol{R}_v$. The RTS smoother estimates of reduced order mode amplitudes, $\hat{\vec{z}}_t$ are used to construct an estimate of the high order states $\hat{\vec{x}}_t \in \mathbb{R}^{n_s}$, with addition of the linear regression fit originally subtracted to centre the data, see (6.27).

To describe the smoothing component of the RTS smoother, recall that it is a backwards pass algorithm. Let the latest discrete time-step with sensing be $k_{sf} \in \mathbb{N}$ and the first discrete time-step with sensing be $k_{s0} \in \mathbb{N}$. Starting at $k = k_{sf} - 1$ and stepping backwards to k_{s0}, the smoothed state estimate is described by the following recursive equations,

$$\hat{\vec{z}}_{k|k_{sf}} = \hat{\vec{z}}_{k|k} + \boldsymbol{\Xi}_k \left(\hat{\vec{z}}_{k+1|k_{sf}} - \hat{\vec{z}}_{k+1|k} \right) \tag{6.36}$$

and

$$\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{k|k_{sf}} = \boldsymbol{P}_{\mathrm{r}}(\vec{p})_{k|k} + \boldsymbol{\Xi}_k \left(\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{k+1|k_{sf}} - \boldsymbol{P}_{\mathrm{r}}(\vec{p})_{k+1|k} \right) \boldsymbol{\Xi}_k^*, \tag{6.37}$$

where

$$\boldsymbol{\Xi}_k = \boldsymbol{P}_{\mathrm{r}}(\vec{p})_{k|k} (\boldsymbol{\Lambda}^{\frac{\delta t}{\Delta t}})^* \boldsymbol{P}_{\mathrm{r}}(\vec{p})^{-1}_{k+1|k}. \tag{6.38}$$

Recall that this state smoothing only occurs prior to an long-term extrapolation or analysis re-run of SCEM, to enhance the accuracy of $\hat{\vec{z}}_t$ (in terms of phase and amplitude). These analysis times are visible in Figure 3.4.

Defining the estimation error for optimisation (Stage 4 → Stage 2)

The reduced order states are modelled as random variables with complex normal distributions, $\vec{z}_t \sim \mathscr{CN}(0, \mathrm{Var}(\mathscr{Z}))$ where $\mathrm{Var}(\cdot)$ is the unbiased sample covariance of a state trajectory. The centred high order states are modelled as real valued normal distributions, $\bar{\vec{x}}_t \sim \mathscr{N}(0, \mathrm{Var}(\bar{\mathscr{X}}))$. In the reduced order system, $\boldsymbol{\Phi}$ is formed of complex conjugate columns, such that it reduces

the imaginary component to zero magnitude when transforming reduced order states to high order states in (6.23). Therefore, a linear transformation of the complex normal distribution $\vec{z}_t \sim \mathscr{CN}(0, \mathrm{Var}(\mathscr{Z}))$ by $\boldsymbol{\Psi}$, results in a normal distribution,

$$\boldsymbol{\Psi}\vec{z}_t \sim \mathscr{N}(0, \boldsymbol{\Psi}\mathrm{Var}(\mathscr{Z})\boldsymbol{\Psi}^*) \approx \bar{\vec{x}} \sim \mathscr{N}(0, \mathrm{Var}(\bar{\mathscr{X}})). \tag{6.39}$$

Note that $\boldsymbol{\Psi}$ is formed from a truncated decomposition and hence is an approximate linear transform. The transform of the covariance of the reduced order complex distribution, to the covariance of the high order real distribution is

$$\mathrm{Var}(\bar{\mathscr{X}}) \approx \boldsymbol{\Psi}\mathrm{Var}(\mathscr{Z})\boldsymbol{\Psi}^*. \tag{6.40}$$

The linear transformation matrix $\boldsymbol{\Psi}$ also transforms the covariance matrix of reduced order state estimation error (see Section 6.1.1), $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k} : \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{C}^{n_z \times n_z}$, to a covariance matrix of state estimation error for current-time high order states. The transformation is described by

$$\boldsymbol{P}(\vec{p})_{t|k} = \boldsymbol{\Psi}\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k}\boldsymbol{\Psi}^*, \tag{6.41}$$

where $\boldsymbol{P}(\vec{p})_{t|k} : \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}^{n_s \times n_s}$ is the estimated error covariance matrix for the current time high order states.

The mean-squared error (MSE) for a state estimation is the diagonal in the error covariance matrix. Hence, the transform of (6.41) is utilised, but only diagonal terms of $\boldsymbol{P}$ are required,

$$\mathrm{MSE}(\hat{\bar{\vec{x}}}_{t|k}) = \mathrm{diag}(\boldsymbol{P}(\vec{p})_{t|k}) \tag{6.42}$$

where $\mathrm{diag}(\cdot)$ extracts the diagonal of a matrix as a column vector. Note the translated MSE, $\mathrm{MSE} : \mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}^{n_s}$, is a real valued column vector containing the MSE for each state. However, (6.42) only describes the estimated MSE of reduced order states (mode amplitudes), translated to the high order system through the mode shapes. This often under-estimates the error in high order state estimates when the identified model is inaccurate, before sensing has corrected the state-trajectory provided by external data. It does not include any description of the error in the mode shapes, an error that emerges from a mismatched state trajectory of the

fluid system. Minimising this mode shape error in (5.1a) by appropriately placing sensors results in more accurate mode shapes. To define an augmented MSE, AMSE : $\mathbb{R}^{n_s} \times \mathbb{R}^{1\times 2n_p} \to \mathbb{R}^{n_s}$, first construct an appended state trajectory reconstruction error matrix, $\widetilde{\bar{\mathscr{X}}} \in \mathbb{R}^{n_s \times n_d}$, where

$$\widetilde{\bar{\mathscr{X}}} = \left[(\bar{\mathscr{X}} - \boldsymbol{USV}^*)^{\circ 2}, \quad (\hat{\vec{x}}_{t|t-\delta t} - \hat{\vec{x}}_{t|t})^{\circ 2} \right]. \tag{6.43}$$

In (6.43) $(\cdot)^{\circ 2}$ is the Hadamard power of 2. The error matrix $\widetilde{\bar{\mathscr{X}}}$ contains the squared error of the reduced order reconstruction of the state trajectory, concatenated with the squared difference between the prior estimated values for the current-time states $\hat{\vec{x}}_{t|t-\delta t} : \mathbb{R}_+ \times \mathbb{R}^{1\times n_p} \to \mathbb{R}^{n_s}$ and the posterior high order state estimate from the reduced order data assimilation, $\hat{\vec{x}}_{t|t} : \mathbb{R}_+ \times \mathbb{R}^{1\times n_p} \to \mathbb{R}^{n_s}$. The augmented MSE is then,

$$\text{AMSE}(\hat{\vec{\bar{x}}}) = \text{MSE}(\hat{\vec{\bar{x}}}) + \max_{\text{row}} \left(\widetilde{\bar{\mathscr{X}}} \right), \tag{6.44}$$

where $\max_{\text{row}}(\cdot)$ extracts the maximum value of each row. Hence, the second term of (6.44) returns the maximum MSE of the reduced order trajectory reconstruction and state estimation. The state trajectory has assimilated prior sensor measurements, and in this case is appended with the MSE of a current-time prior and the posterior state estimate. Therefore the augmented MSE is likely largest where a sensor reading has significantly differed from a poor prior state estimation. This provides incentive for a sensor to examine a well-determined mode amplitude, but badly-determined mode shape.

The variance sources $E_{\text{KF}_x}(\vec{p})$ and $E_{\text{KF}_y}(\vec{p})$, which describe the assimilation error in horizontal and vertical fluid velocity, are utilised in the uncertainty description (4.86) to inform the sensor optimisation of poorly resolved or estimated regions of the flow field. The variance sources $E_{\text{KF}_x}(\vec{p})$ and $E_{\text{KF}_y}(\vec{p})$ are the subsets of the augmented MSE corresponding to the horizontal and vertical velocity augmented MSE at a specific spatio-temporal location. The term $E_{\text{KF}}(x_i, y_j, t, \vec{p})$ in (5.2), is the magnitude of the vector $[E_{\text{KF}_x}(\vec{p}), E_{\text{KF}_y}(\vec{p})] \in \mathbb{R}^2$, min-max normalised across the spatial domain. In the adjoint method, the derivatives of $E_{\text{KF}_x}(\vec{p})$ and $E_{\text{KF}_y}(\vec{p})$ are required. In (6.44), it is assumed that $\max_{\text{row}} \left(\widetilde{\bar{\mathscr{X}}} \right)$ is constant, so the derivative reduces to the derivative of the time-varying Kalman filter estimation-error covariance matrix update equation. This derivative is described in Appendix A.

The weighting matrix for the BIRSVD in Section 6.1, seeks to mitigate the creation of reduced order mode shapes that do not represent true flow field characteristics. Poor mode shapes can occur due to erroneous states in the high order state trajectory, that can be present due to inaccurate external data. The weighting matrix is described in a block-wise fashion, for each centred state vector $\bar{\vec{x}}_t$ in $\bar{\mathscr{H}}_r$,

$$\boldsymbol{W}(\bar{\vec{x}}_t) = \text{median}\left(\left[0, R_{\text{sens}}\,\text{AMSE}(\bar{\vec{x}}_t)^{\circ -1}, 1\right]\right). \tag{6.45}$$

In (6.45), $R_{\text{sens}} \in \mathbb{R}_+$ is the sensors mean squared measurement error for homogeneous sensors, $(\cdot)^{\circ -1}$ is the Hadamard inverse and $\text{AMSE}(\bar{\vec{x}}_t) \in \mathbb{R}^{n_s}$ is the estimated augmented MSE of that state vector within the Hankel state trajectory. The median$(\cdot)$ function returns the median of the three values in (6.45), limiting $R_{\text{sens}}\,\text{AMSE}(\bar{\vec{x}}_t)^{\circ -1}$ to the interval $[0, 1]$. Hence the weightings $\boldsymbol{W}(\bar{\vec{x}}_t)$ are the inverse of the augmented MSE of each state, as a proportion of the sensor MSE.

Past and future estimation (Stage 4)

The DMD model is used to extrapolate the fluid flow before sensors have arrived. Beginning at the closest time before the oil release that matches the time of day of the present DMD model and mode amplitude estimate, the reduced order model steps forward in time, from the current condition and thus exploits the periodicity of tidal cycles but with amplitude growth or decay from the linear-regression fit. The modal trajectory is calculated up to the activation of sensors. The estimated modal amplitudes reconstruct a fluid flow trajectory that is utilised in a re-run of the oil simulation before sensors arrive. The reduced order model is also used to augment the external data used as inputs for SCEM. As SCEM predicts towards the future, the RTS smoother utilises both external data and DMD predicted flow as measurements to correct the fluid flow.

6.2 Time varying reduced order system and the RTS smoother

The reduced order system is re-identified whenever new sensor measurements are available, therefore it is necessary to transform the RTS smoother prior estimations of states and error

covariance to the new mode-shapes of the system before correction. This accounts for possible changes in mode shape (basis functions) from new data that has become available. The same transformation of stored state-estimates and covariance matrices to the most-recent reduced order system is required when stepping backwards in time with the RTS smoother. This section uses time subscripts, hence time dependence is defined explicitly. Define two mode matrices $\boldsymbol{\Phi}_{k-1} : \mathbb{N}^{n_t} \to \mathbb{C}^{n_s \times n_z}$ and $\boldsymbol{\Phi}_k : \mathbb{N}^{n_t} \to \mathbb{C}^{n_s \times n_z}$ for the previous-time-step and current-time-step reduced order systems respectively, as functions of a changing state-trajectory. Furthermore, let $\hat{\vec{z}}_{t|k-1} : \mathbb{R}_+ \to \mathbb{C}^{n_z}$ and $\hat{\vec{z}}_{t|k} : \mathbb{R}_+ \to \mathbb{C}^{n_z}$ be the previous-time-step and current-time-step prior estimation of reduced order states, then note $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k-1}$ and $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1} : \mathbb{R}_+ \times \mathbb{R}^{2n_p} \to \mathbb{C}^{n_z \times n_z}$ as the corresponding error covariance matrices. The extended subscript $t|k|k-1$ reflects that this is the estimated error covariance for time t, using the DMD system at k, but the available information at $k-1$. This is still a prior Kalman variable. Handling the reduced order states is straight-forward,

$$\hat{\vec{z}}_{t|k} = \boldsymbol{\Phi}_k^{+}(\boldsymbol{\Phi}_{k-1}\hat{\vec{z}}_{t|k-1}), \tag{6.46}$$

where $^+$ is the Moore-Penrose inverse. However translating the error covariance is more involved. Transforming a complex reduced order covariance to a high order covariance is described by

$$\boldsymbol{P}_{t|k-1} = \boldsymbol{\Phi}_{k-1}\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k-1}\boldsymbol{\Phi}_{k-1}^{*}, \tag{6.47}$$

where $\boldsymbol{P}_{t|k-1} : \mathbb{R}_+ \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_s \times n_s}$ is the error covariance of fluid states in the full system. For the error covariance, the problem statement is solving for $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1}$, the current-time error covariance for the most recently identified reduced order system, in

$$\boldsymbol{P}_{t|k|k-1} = \boldsymbol{\Phi}_k\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1}\boldsymbol{\Phi}_k^{*}. \tag{6.48}$$

The term, $\boldsymbol{P}_{t|k|k-1} : \mathbb{R}_+ \times \mathbb{R}^{n_p} \to \mathbb{R}^{n_s \times n_s}$ is the full-order estimated error covariance for time t, using the DMD system at k, with the available information at $k-1$ before the Kalman filter update at k. Though (6.48) is an over-determined system, recall that $\boldsymbol{\Phi}_k$ and $\boldsymbol{\Phi}_k^{*}$ are non-square, low-rank and complex valued, and the solution is required to be unique and positive semi-definite as $\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1}$ and $\boldsymbol{P}_{t|k|k-1}$ are both covariance matrices. Theoretically, a Moore-Penrose inverse based solution should provide a positive semi-definite solution as both

$\boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1}$ and $\boldsymbol{P}_{t|k|k-1}$ are strictly positive semi definite matrices. However, under the influence of rounding error induced numerical instability or inconsistency where $\boldsymbol{P}_{t|k|k-1} \neq \boldsymbol{P}^*_{t|k|k-1}$ in implementation, a solution method with a guarantee of positive semi-definiteness prevents instability of the RTS smoother. The general, inconsistency capable method described in Section 3 of Hua and Lancaster 1996 is implemented. To summarise this approach, first let $\boldsymbol{P}_{\mathrm{r}} = \boldsymbol{P}_{\mathrm{r}}(\vec{p})_{t|k|k-1}$, $\boldsymbol{Q} = \boldsymbol{\Phi}^*_k, \in \mathbb{C}^{m\times n}$ of $m, n \in \mathbb{N}$ dimensions, and $r = \mathrm{rank}(\boldsymbol{Q}) \in \mathbb{N}$. Consider the solution to (6.48) as the solving $\boldsymbol{P}_{\mathrm{r}}$ in the following minimisation,

$$\min_{\boldsymbol{P}_{\mathrm{r}}} \left\| \boldsymbol{Q}^* \boldsymbol{P}_{\mathrm{r}} \boldsymbol{Q} - \boldsymbol{P}_{t|k|k-1} \right\|_F, \tag{6.49}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Perform the full SVD of $\boldsymbol{Q} = \boldsymbol{U}_{\boldsymbol{Q}} \boldsymbol{\Sigma} \boldsymbol{V}^*_{\boldsymbol{Q}}$ and assemble the partitioned form,

$$\boldsymbol{Q} = \boldsymbol{U}_{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{V}^*_{\boldsymbol{Q}}, \tag{6.50}$$

where the unitary matrices are split, $\boldsymbol{U}_{\boldsymbol{Q}} = [\boldsymbol{U}_{\boldsymbol{Q}_1}, \boldsymbol{U}_{\boldsymbol{Q}_2}] \in \mathbb{C}^{n\times n}$, $\boldsymbol{V}_{\boldsymbol{Q}} = [\boldsymbol{V}_{\boldsymbol{Q}_1}, \boldsymbol{V}_{\boldsymbol{Q}_2}] \in \mathbb{C}^{m\times m}$, the subsets are $\boldsymbol{U}_{\boldsymbol{Q}_1} \in \mathbb{C}^{n\times r}$, $\boldsymbol{V}_{\boldsymbol{Q}_1} \in \mathbb{C}^{m\times r}$. The subsets $\boldsymbol{U}_{\boldsymbol{Q}_2}$ and $\boldsymbol{V}_{\boldsymbol{Q}_2}$ are the remaining components of $\boldsymbol{U}_{\boldsymbol{Q}}$ and $\boldsymbol{V}_{\boldsymbol{Q}}$. Substitute the partitioned form of $\boldsymbol{Q}$ into (6.49),

$$\min_{\boldsymbol{P}_{\mathrm{r}}} \left\| \left(\boldsymbol{U}_{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{V}^*_{\boldsymbol{Q}} \right)^* \boldsymbol{P}_{\mathrm{r}} \boldsymbol{U}_{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{V}^*_{\boldsymbol{Q}} - \boldsymbol{P}_t \right\|_F. \tag{6.51}$$

By expansion of the conjugate transpose and through further partitioning and rearranging, (6.51) becomes

$$\min_{\boldsymbol{P}_{\mathrm{r}}} \left\| \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}^*_{\boldsymbol{Q}} \boldsymbol{P}_{\mathrm{r}} \boldsymbol{U}_{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} - \begin{bmatrix} \boldsymbol{V}^*_{\boldsymbol{Q}_1} \boldsymbol{P}_t \boldsymbol{V}_{\boldsymbol{Q}_1} & 0 \\ 0 & 0 \end{bmatrix} \right\|_F. \tag{6.52}$$

Through inspection of (6.52), the minimising solution is of the form,

$$\boldsymbol{P}_{\mathrm{r}} = \boldsymbol{U}_{\boldsymbol{Q}} \begin{bmatrix} \boldsymbol{P}_{\mathrm{r}0} & 0 \\ 0 & 0 \end{bmatrix} \boldsymbol{U}^*_{\boldsymbol{Q}}, \tag{6.53}$$

where $\boldsymbol{P}_{\mathrm{r}0} = \boldsymbol{\Sigma}^{-1}\boldsymbol{V}_{\boldsymbol{Q}_1}^{*}\boldsymbol{P}_t\boldsymbol{V}_{\boldsymbol{Q}_1}\boldsymbol{\Sigma}^{-1}$ is the natural, but non-positive-semi-definite enforced solution. To constrain $\boldsymbol{P}_{\mathrm{r}}$ to a positive semi definite solution, it is only necessary to enforce symmetry as $\boldsymbol{P}_t$ is also positive semi definite.

It is known that the nearest symmetric matrix to (any) the given matrix $\boldsymbol{P}_{\mathrm{r}0}$ in the F-norm is $\frac{1}{2}(\boldsymbol{P}_{\mathrm{r}0}^{*}+\boldsymbol{P}_{\mathrm{r}0})$. Applying this to $\boldsymbol{P}_{\mathrm{r}0} = \boldsymbol{\Sigma}^{-1}\boldsymbol{V}_{\boldsymbol{Q}_1}^{*}\boldsymbol{P}_t\boldsymbol{V}_{\boldsymbol{Q}_1}\boldsymbol{\Sigma}^{-1}$ for a minimising symmetric solution X_s gives,

$$\boldsymbol{P}_{\mathrm{r}s} = \frac{1}{2}\boldsymbol{\Sigma}^{-1}((\boldsymbol{V}_{\boldsymbol{Q}_1}^{*}\boldsymbol{P}_t\boldsymbol{V}_{\boldsymbol{Q}_1})^{*} + (\boldsymbol{V}_{\boldsymbol{Q}_1}^{*}\boldsymbol{P}_t\boldsymbol{V}_{\boldsymbol{Q}_1}))\boldsymbol{\Sigma}^{-1}. \tag{6.54}$$

To demonstrate the positive semi-definite enforcement of (6.54), consider that if $\boldsymbol{P}_t$ is positive semi-definite, then $\boldsymbol{V}_{\boldsymbol{Q}_1}^{*}\boldsymbol{P}_t\boldsymbol{V}_{\boldsymbol{Q}_1}$ must also be positive semi-definite. Therefore, $\boldsymbol{P}_{\mathrm{r}s}$ must also be positive semi-definite. The complete solution for $\boldsymbol{P}_{\mathrm{r}}$, when $\boldsymbol{P}_{\mathrm{r}}$ is constrained to be symmetric and therefore positive semi-definite, is

$$\boldsymbol{P}_{\mathrm{r}} = \boldsymbol{U}_{\boldsymbol{Q}}\begin{bmatrix}\boldsymbol{P}_{\mathrm{r}s} & 0\\ 0 & 0\end{bmatrix}\boldsymbol{U}_{\boldsymbol{Q}}^{*}. \tag{6.55}$$

Chapter 7

Evaluation

This chapter utilises the monitoring framework in three scenarios, all in the coastal waters near Hong Kong, modified from the 2019 Aulac Fortune spill. The same location is used for clarity, though the SCEM model has been utilised worldwide, for the 2018 Sanchi spill near China, the 2019 MV Solomon Trader near the Solomon Islands, the 2019 Bay of Biscay spill and others mentioned in Chapter 8. The scenarios have benchmark spill results using TideTech ocean data, which includes the tidal dynamics of the Zhujiang river, while the monitoring framework has initial data from the Global Forecasting System, which does not include tidal dynamics. Sensing must therefore correct the lack of tidal flow through measurement, at locations determined by one-step-ahead solving of the uncertainty minimisation when using the monitoring framework. The first scenario discussed is a forward monitoring problem. Multiple sensors are activated from the spill source, when the spill occurs, and monitor the spill forward in time. Sensing ends and the spill is predicted into the future. The monitoring framework is compared with ladder path sensing with value replacement feedback of environment properties. The value replacement feedback represents the inability of a standard oil model relying upon external fluid data to properly assimilate data into the fluid flow without an internal model. This first scenario represents a well-equipped response team immediately responding to an incident.
The second scenario is considerably more complex. A single sensor arrives 12 hours late to the spill. The spill source location is known but the spill location is not, and the sensors are tasked with mapping and monitoring the spill, and determining the spill trajectory prior to sensor arrival. This represents a standard realistic scenario, where an initial response team with reduced resources responds to an incident in a remote location. The monitoring framework is

compared with ladder path sensing, which also uses the monitoring framework, but without the optimisation determined sensor pathing. There is also some discussion here about the effect of additional sensors with each pathing method.

In the final scenario, a spill is detected from an unknown source, 10 hours after the spill has occurred. The initial spill detection contains false positive locations and false negative locations, though it has some overlap with the benchmark spill and a completely disparate false spill. This is representative of synthetic aperture radar imagery in a coastal environment with sheltered areas, where both wind speeds and oil presence can cause great variation in ocean surface roughness. This surface roughness, or lack-of, can lead to false positive and negatives of oil presence. In the scenario, 2 hours after the spill is detected, a single sensor arrives and is required to map and monitor the spill, determine the source location through spill backtracking, and establish the trajectory of the spill prior to sensor arrival. In this scenario it is assumed the sensor can determine the spill age, or time-in-water, to within an hour. This may be beyond the capability of current mobile sensors, but when combined with more human information (crews noticing oil in the water, people on the beach noticing oil), it is not unreasonable to expect to establish oil age with some accuracy.

7.1 Monitoring of an oil spill

This test case is a hypothetical 100 barrel spill of light crude oil near Hong Kong at 1900 hours on the 8th of January 2019. To provide measurements, 4 mobile sensors capable of measuring oil particles, wind and current velocities arrive 1 hour after the leak begins and stay for 14 hours, de-activating at 0900 hours on the 9th January. The sensors are speed limited to 60 miles-per-hour, model guided sensors are measuring only in 15-minute intervals at point locations while industry sensors have been given the capability to continuously measure while following waypoints. Spill prediction continues to 1900 hours on the 9th of January 2019.

The real simulation, from which sensors measure, utilises data from the Global-Forecast-System (GFS) for wind velocities and Tidetech data for current velocities that include both global circulation currents and tidal flow. The test simulations use the same wind velocity data, but instead use GFS current data that does not include tidal flow which is critical for spill prediction in this region, at the mouth of the Zhujiang river.

Industry pathing prescribes a ladder flight path (IPIECA and IOGP 2016; ITOPF 2011b) that covers where oil is predicted to be by the model, with a 10% overlap selected here. The path plan is split up into sections, one for each sensor, with spacing sufficient to ensure no oil can be missed during flight. Detected oil or clear areas are updated in the model, but velocity and wave spectrum data is only utilised as a value-replacement in the model to reflect the inability of traditional models to utilise measured velocity data in the same manner as SCEM. Figure 7.2 also includes the error of industry pathing with no velocity feedback, to represent a simple model incapable of modifying external flow data. The ladder flight path is updated every hour to enable sensors to respond to measured oil, with sensors repeating the path at maximum speed for the highest frequency of measurements along the path.

7.1.1 Analysis of results

The simulation using industry pathing and oil particle updates was accurate before sensors were deactivated (see Figure 7.2), but once sensors deactivate and the model loses high frequency updates on particle positions, the inaccurate velocity field causes the main body of the industry spill to drift 5 km to the North East of the real spill. For the model-based method, after sensors were deactivated the prediction model of SCEM had been sufficiently modified by measured data to produce a more accurate velocity field and maintain accuracy when advecting the particles, with the main body of particles within the real spill location even 3 hours after sensors had deactivated.

Analysing the oil presence error in Figure 7.2, model guided sensing with state estimation has a 30% reduction in the area of incorrect oil presence from the industry method with feedback, or 50% better than the industry method with only oil information feedback, both when sensors are active and after sensor removal at 0900 hours on the 9th January. After sensor deactivation, the industry method rapidly becomes less accurate as it is still utilising the incorrect input data to predict the spill drift. The model guided sensors have partially corrected the SCEM fluid model to include tidal flow and so while this prediction also loses accuracy after sensors are deactivated, it continues to perform better than using no sensors and the incorrect input data, unlike the industry method. In both Figure 7.2 and Figure 7.3 the error of the model guided sensors is slightly above that of industry sensors with value replacement feedback. This is due

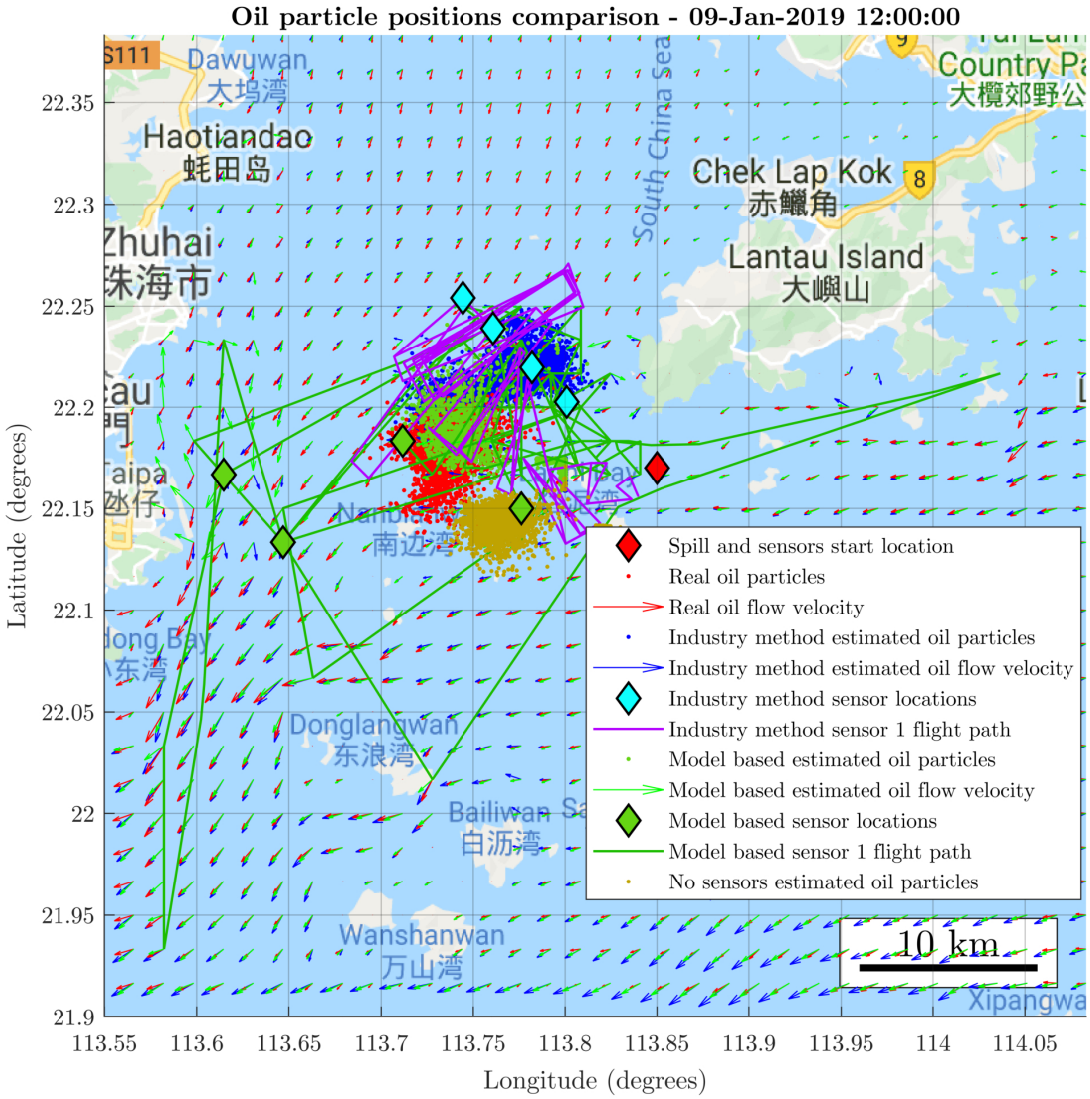


Figure 7.1: A comparison of the simulations at 1200 hours on the 9th January 2019, 15 hours after the initial leak at the indicated spill location, and 3 hours after the 4 sensors have deactivated though their final positions are displayed. Real oil particle locations are displayed in red and are advected by the red velocity field. The simulation using incorrect input data and no sensors has particle locations displayed in gold and the main body of this spill is not within the real spill body. The simulation using the industry method of sensor pathing and feedback, has particle positions and velocity field displayed in dark blue. The main body of the industry spill is 5 km to the North East of the real spill. The simulation using model based sensor behaviour and state estimation has the main body of particles (green) within the body of the real spill. Note how the flight path of industry sensor 1 (purple) concentrates over the predicted spill location in an expanding ladder path from the spills initial position, while the flight path of model based sensor 1 (dark green) also flies to crucial velocity measuring locations both up and downstream of the spill, before returning to check the spill. Map data ©2020 Google.

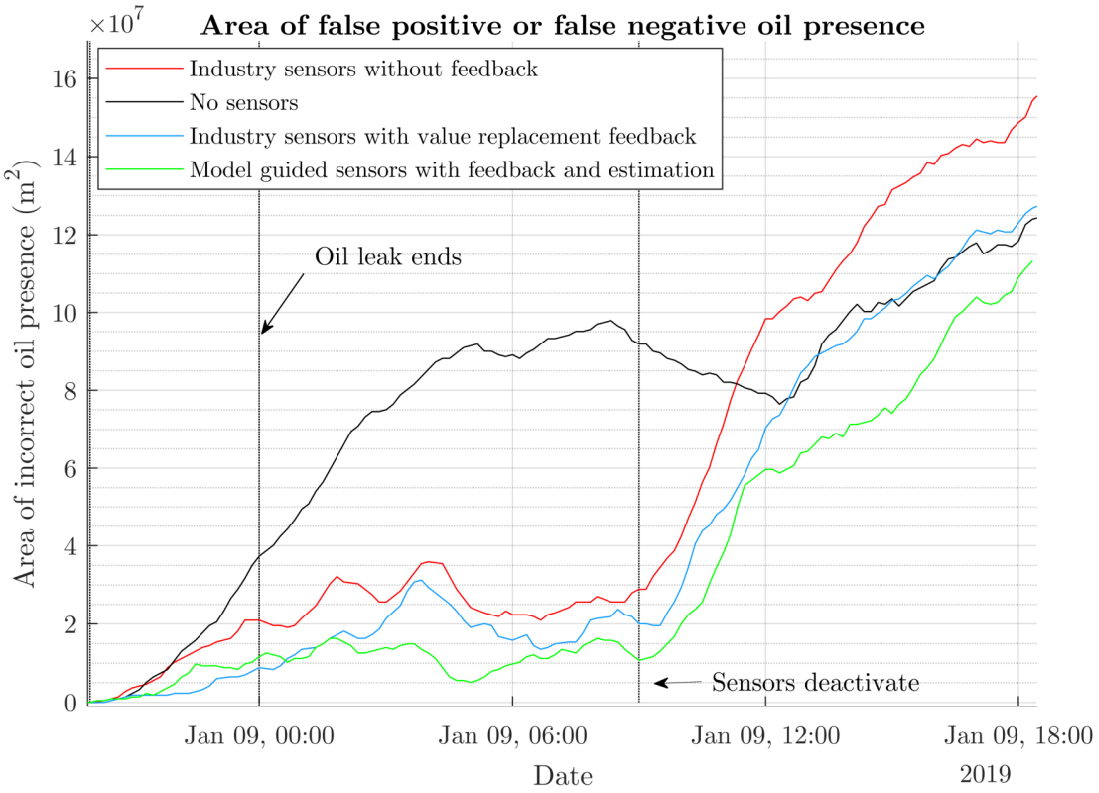


Figure 7.2: A comparison of the oil presence error of the simulations. Note how all sensor approaches reduce error by 70% while sensors are active, with the model guided sensors being approximately 60% as erroneous as industry standard pathing and continuing to have less error after sensing stops.

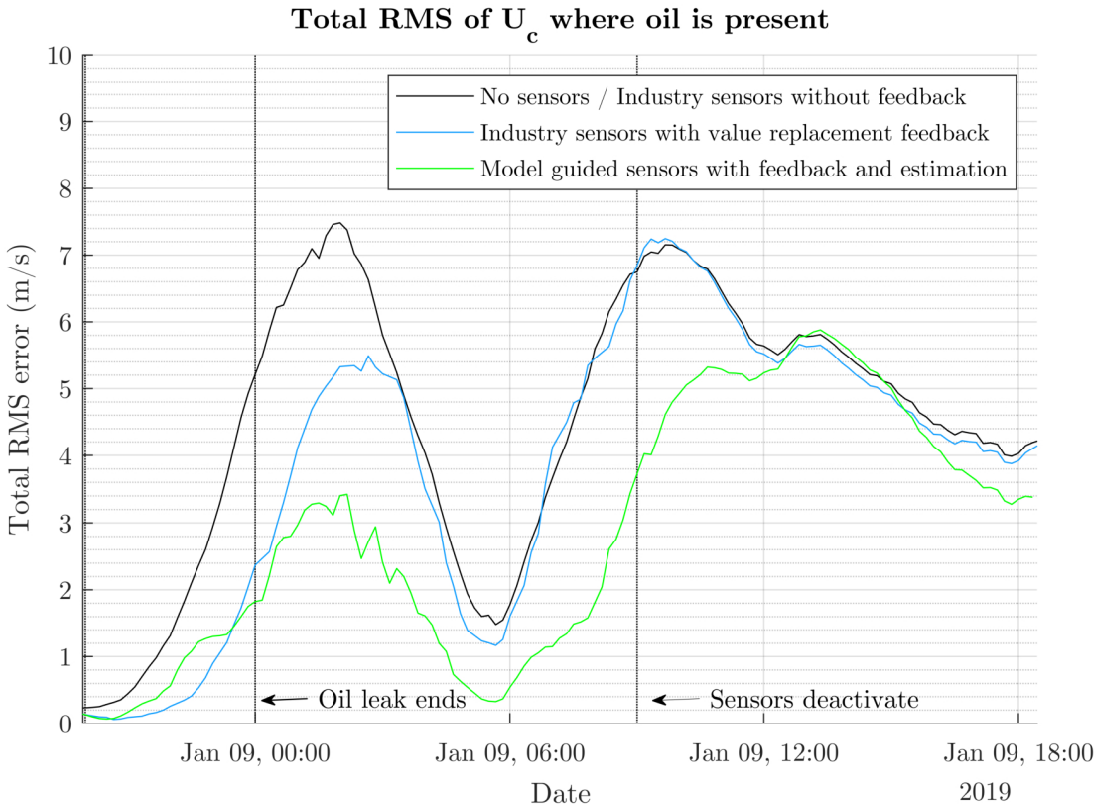


Figure 7.3: A comparison of the RMS error of ocean flow velocity where oil is present. Note the 30% to 50% in reduction in error the model guided sensors with feedback and estimation displays over the industry sensors using value replacement, and the continuation of less error after sensing stops.

to the sensors flying further from the spill to measure crucial flow regions and temporarily compromising their update-rate of states local to the spill, though is important for reducing the long-term error.

Although the area covered by the model based flight path of sensor 1 is much greater, as seen in Figure 7.1, the distance moved is actually the same or less than the sensor 1 using the industry method: In the industry method, the sensors fly the ladder path at their maximum speed for the whole time sensors are active, repeating the path and measurements as often as possible before the next ladder path is generated as the update rate is crucial to the accuracy of the industry method. Meanwhile, the model-based method will relocate sensors to optimal positions that may or may not require flying at maximum speed.

7.1.2 Conclusion

In this section the framework has demonstrated improvement in present-time monitoring on a test case and a capability for online model adjustment to better predict future spill dynamics.

7.2 Monitoring and analysis of an oil spill

A test case is formed from a high fidelity simulation of the 2019 Aulac Fortune tanker explosion, with initial conditions modified to better examine the monitoring framework. A 100 barrel spill of light crude oil near Hong Kong, began at 19:00 on the 8th of January and the leak continued until 00:00 on the 9th January. The simulation providing the benchmark results utilises data from the Global-Forecast-System (GFS) for wind velocities and Tidetech data for ocean current velocities that includes tidal flow. The test simulations use the same wind velocity data as the benchmark simulation, but instead use GFS ocean circulation current data. The GFS data lacks the tidal flow within the Tidetech data, which is critical for spill prediction in this region, at the mouth of the Zhujiang river. Hence the monitoring framework will be required to resolve the additional tidal flow component. This is representative of a common real-life simulation case where global circulation or mean-flow data is available, but local tidal data is unavailable. In this region, the Hong Kong Tidal Stream Prediction System available from the Hong Kong Hydrographic Office Marine Department does not provide tidal flow data to the west of Lantau Island. The difficulty of tidal estimation is increased, as the tide flow is mixed semi-diurnal (first

and second tides of the day differ) and in January is moving towards a spring tide, increasing in magnitude by approximately 5% per day.

A mobile sensor that measures oil concentration, wind and ocean velocities in point locations arrives 12 hours after the leak begins, and remains active for 24 hours, de-activating at 07:00 on the 10th January. Spill prediction continues to 08:00 on the 10th of January.

7.2.1 Experiment setup

This paper compares the benchmark results with test simulations utilising;

(A) No sensors.

(B) A constantly measuring sensor using industry standard ladder pathing and the monitoring framework but with simple value-replacement data assimilation, as would be available to external-data reliant oil models.

(C) A constantly measuring sensor using industry standard ladder pathing and the monitoring framework. This enables evaluation of full pathing autonomy with the uncertainty method, compared to the common ladder pathing.

(D) A constantly measuring sensor using optimisation pathing and the monitoring framework (the approach here).

(E) A 15-minute interval measuring sensor using optimisation pathing and the monitoring framework. The sensor measurement frequency represents the time taken to survey the area within a grid cell, with the reduced sensor swathe of sensors mounted on small autonomous aircraft.

(F) 1/2/3/4 sensors using the ladder pathing and optimisation pathing, to investigate the utilisation of extra sensors.

Only experiments (A), (C) and (D) are compared in figures, with (F) also displayed in Table 7.2. However, the results of (B) and (E) are discussed. This selection of tests enables evaluation of the monitoring framework, the optimisation determined sensor trajectory and the analysis/hindcast effectiveness.

A speed constraint limits the sensor to 60 miles-per-hour. Sensors following a ladder flight path (IPIECA and IOGP 2016; ITOPF 2011b), use spacing determined by the sensor swathe to ensure no oil can be missed during flight. The ladder path is updated every 30 minutes to respond to new information. Industry sensors always fly at their maximum speed and sensors repeat their route should they reach the end before a route update, offering the highest frequency of information acquisition across their path. The ladder plan overlaps where the oil is predicted to be by the model, with 20% extra length and width and is split into equal sections, with each assigned to a sensor if there are multiple.

7.2.2 Performance measures

To compare test results, a performance scalar $e_p : \mathbb{R}_+ \to [0, 1]$ is the equally weighted sum of:

- The distance between spill centres normalised by the diagonal dimension of the domain surface. The spill centre is defined as the mean location of non-beached particles. This definition avoids ambiguity in spill geometry definitions, such as choosing the division between two distinct but overlapping spill slicks.

- The volume distribution error between the spills, after the test spill has been moved to the mean location of the benchmark spill or remaining in its original position, whichever is smaller. Beached particles are not relocated. This is normalised by the total volume of both spills.

Note this includes error in spill position and volume distribution across the domain. The performance measure has a maximum value of 1, indicating an estimated spill entirely in the wrong location (the opposite side of the surface domain), with an incorrect spill shape resulting in no-overlap of spills, even when adjusted such that the mean positions of the benchmark and estimate spill match. A low score indicates an accurate estimation, with a score of 0 indicating an identical estimated spill to benchmark spill. The mean value of e_p over a time period is denoted $\bar{e}_p \in [0, 1]$. Simulations compare their forward estimation and analyses over the first 24 hours of the spill. This time window is equally divided between 12 hours analysis-only and 12 hours sensors-active time and will be used to compare discovery and mapping speed of the spill, the forward monitoring effectiveness and the analysis performance prior to sensor arrival.

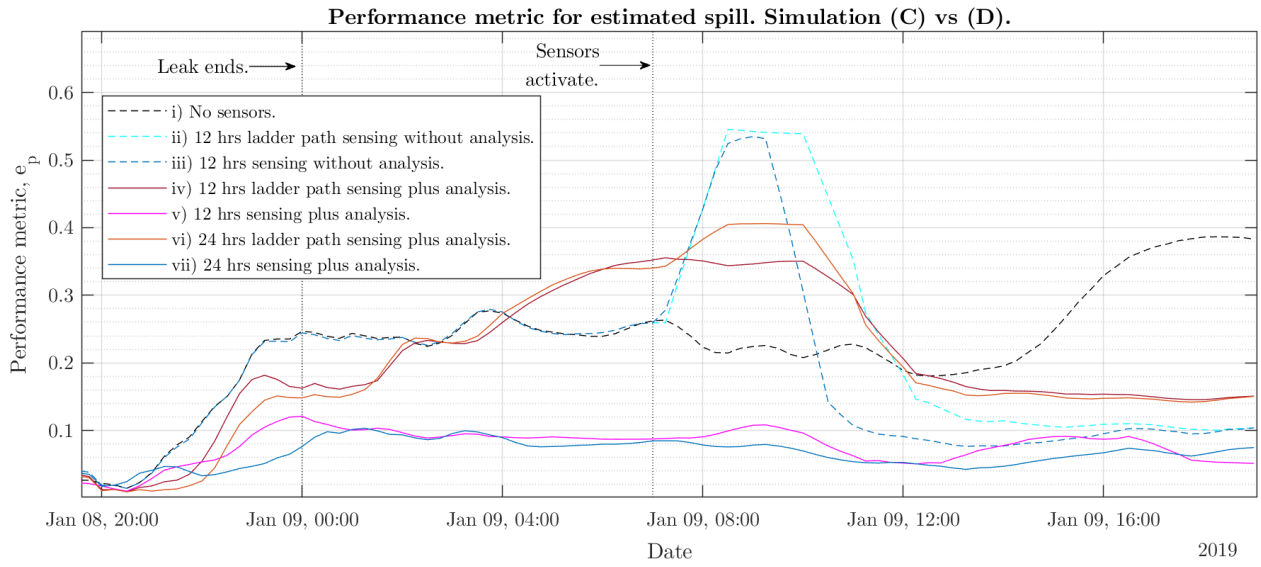
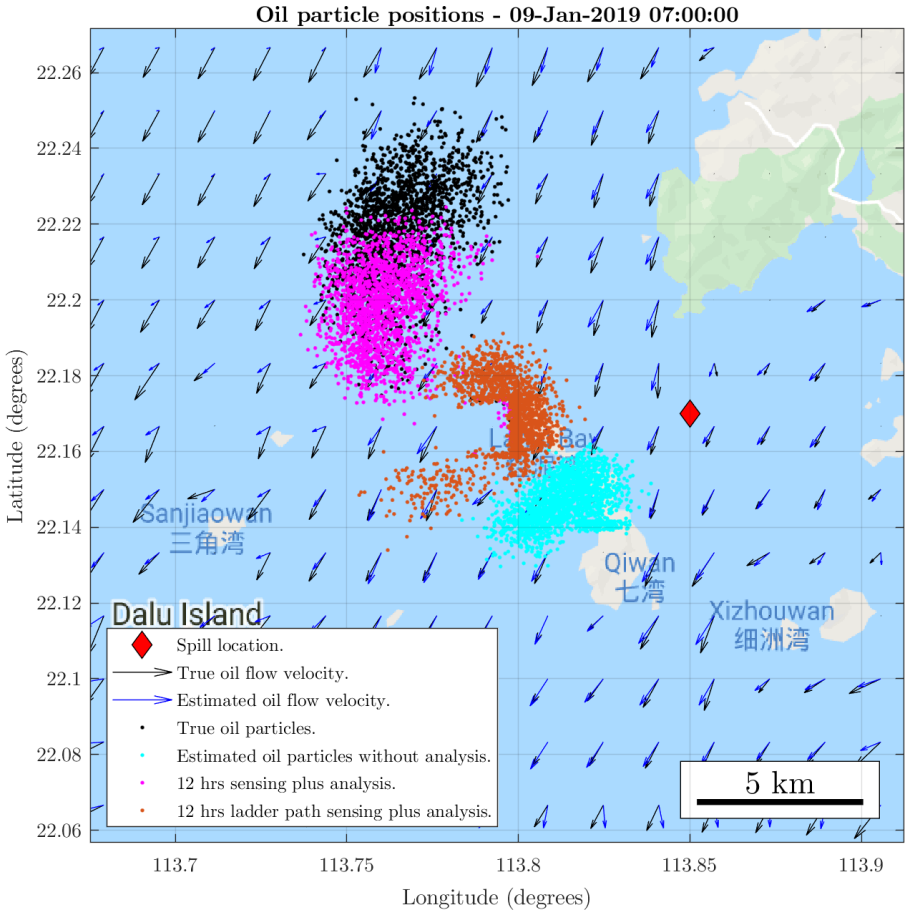
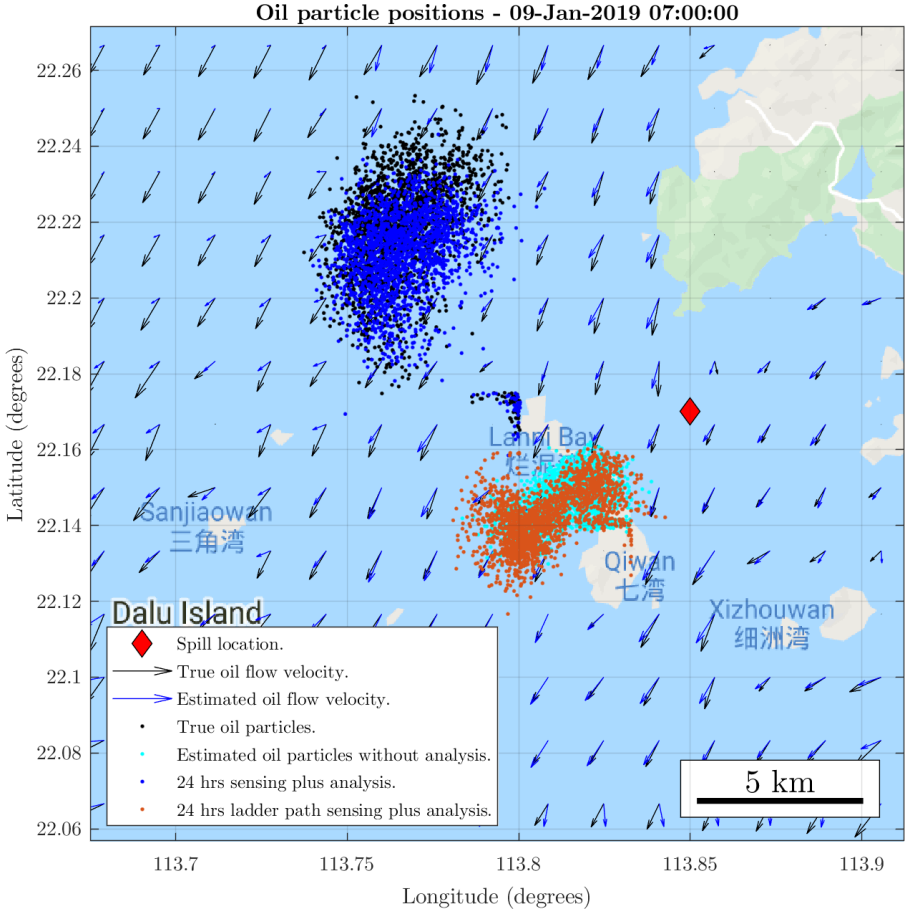


Figure 7.4: This figure displays the performance metric for the first 24 hours of the spill for the test simulations (C) and (D). Both strategies without analysis or sensors are identical until sensor activation. Then, as the sensors deploy and ascertain that the pre-analysis estimated spill is inaccurate and remove the cyan particles in Figure 7.5, the error increases immediately after 07:00 on Jan 9th, as there is now an estimation of no spill in the domain. The optimisation pathed sensing (iii) discovers the benchmark spill at approximately 10:30 on the 9th January, with a corresponding reduction in error as this spill is mapped. The ladder path sensing (ii) discovers the benchmark spill at 09:00, as the strategy resorts to a large area survey once the original estimate is removed. With the spill mapped, the ladder pathing method outperforms the no-analysis optimisation pathing by around 15%. Note that following analysis after 12 hours of sensing, the optimisation pathing (v) and monitoring framework reduces average error over the time domain by approximately 50% and by a further 20% after 24 hours of sensing (vii), while the ladder pathing method (iv and vi) does not produce an accurate analysis. Note that although 12 hours optimisation pathing analysis outperforms the 24 hours at the end of the displayed time window, this is reversed later in time and the practical difference in location and distribution is negligible.



7.5.a: Comparison of oil particles after 12 hours of sensing.



7.5.b: Comparison of oil particles after 24 hours of sensing.

Figure 7.5 *(previous page)*: A comparison of oil particle positions at 07:00 on the 9th January 2019, 12 hours after the leak at the indicated location, before sensor activation. Black dots are the benchmark oil particle positions, while the cyan dots display the estimated location of spill particles using incorrect fluid data without analysis. Magenta dots display the estimated location of spill particles at 07:00, after 12 hours of sensing with analysis. The position error using optimisation determined pathing is reduced from the estimated spill being 7 km to the south-east of the benchmark spill, to the estimated spill being 2 km to the south of the benchmark spill. Blue dots display the estimated location of spill particles after 24 hours of optimisation pathed sensing with analysis, with negligible error in spill location but small error in spatial distribution. The orange dots in the left figure are the estimated location of spill particles after 12 hours of ladder path sensing with analysis and the spill position error has reduced, with the estimated spill being 4 km to the south-east of the benchmark spill, but with a large error in spatial distribution. The orange dots in the right figure are the estimated spill position after 24 hours of ladder path sensing, with analysis, and they have no improvement upon the original estimation. Map data ©2020 Google.

Performance $\bar{e}_p$ vs number of sensors n_p				
n_p	1	2	3	4
Sensors + analysis	0.21	0.07	0.05	0.04
LP Sensors + analysis	0.24	0.17	0.19	0.10

Table 7.2: This table displays the scalar performance metric (lower is better) for test simulations using different numbers of sensors with analysis and ladder path (LP) sensors with analysis. The optimisation pathing method improves upon the ladder pathing method by an average of 50%, and improves consistently with the addition of new sensors, compared to the irregular behaviour of ladder pathed sensors.

7.2.3 Discussion

Discussion will follow a chronological path through the spill release and sensing, then analysis. Much of the description of results is contained in figure captions and this section will focus upon insight and explanation.

Pre-sensors, without analysis

Referring to Figure 7.4, spill release begins 19:00 on the 8th January and before sensors activate there is negligible oil presence error difference between no sensing and any sensing method without analysis. They are not identical due to the random processes in oil behaviour. The similarity is expected because they are using the same external ocean data without tidal flow and no correction, and is observed in the lack of difference in error of Figure 7.4.

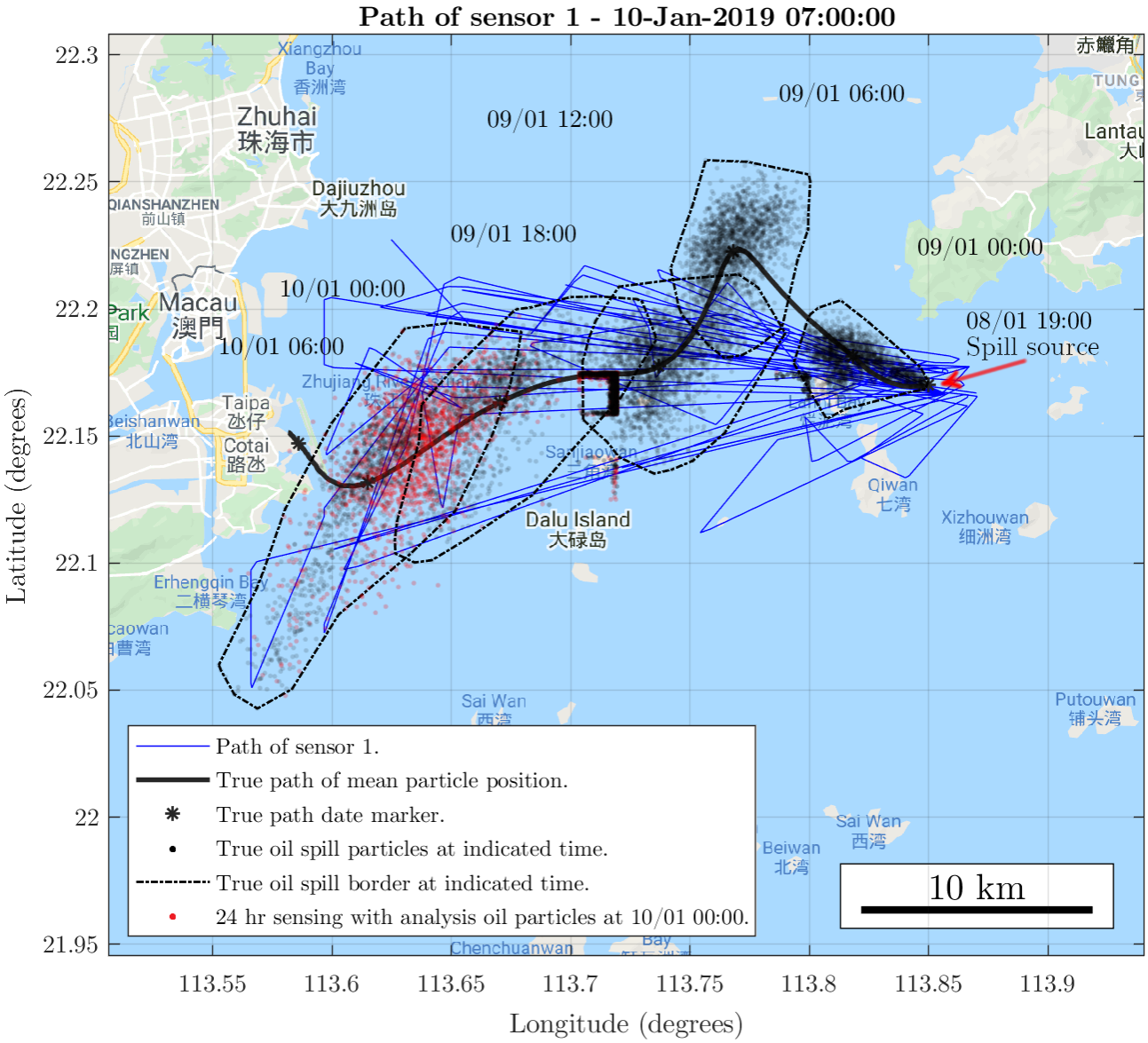


Figure 7.6: The optimisation determined sensor trajectory, for sensor 1. Note the number of sensor flights made away from the spill particles and their mean position, to measure crucial regions of fluid flow and around the spill source. The sensor often alternates between a source measurement, a current-time spill measurement and an up or down-stream measurement en-route between the source and current-time spill. Map data ©2020 Google.

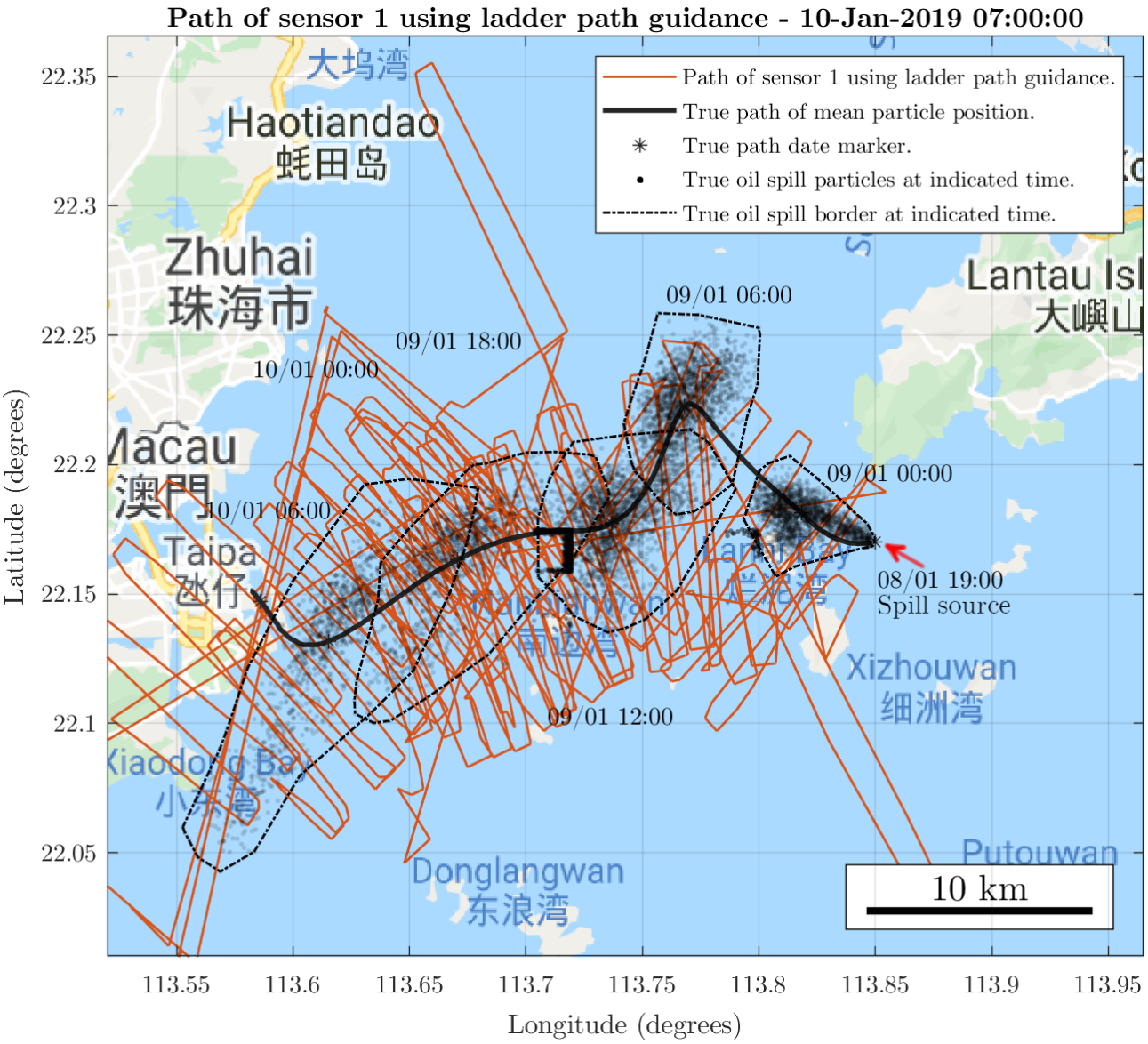


Figure 7.7: A depiction of the ladder path sensor trajectory for sensor 1 is in orange. Note the coverage of the spill particles and mean position and structured flight path. The unusually long flight paths to the north and south are from the large area survey conducted to discover the benchmark spill after removal of the poor initial estimate. The sensor does not return to the prior positions of the spill to form an accurate analysis. Map data ©2020 Google.

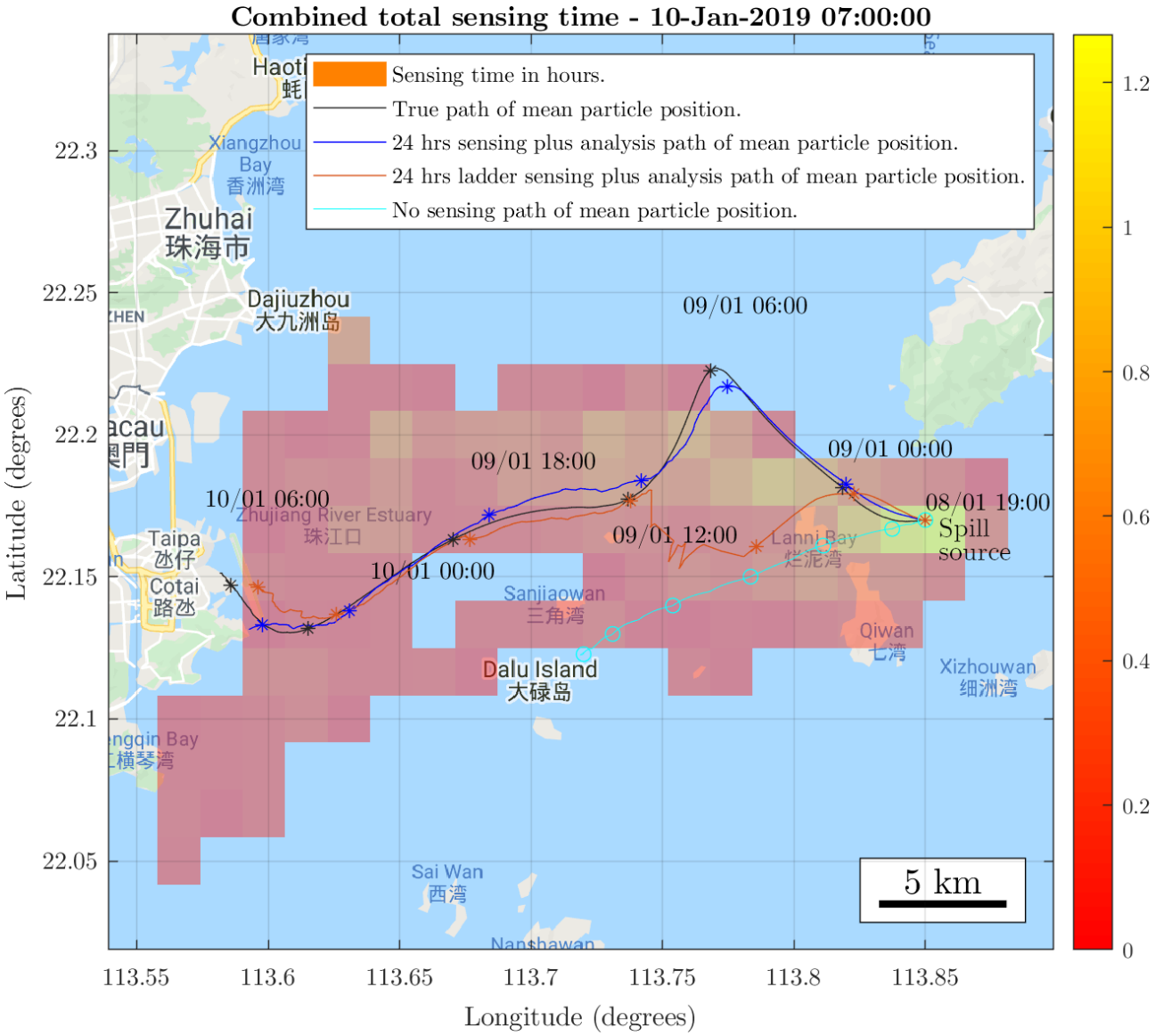


Figure 7.8: A heat-map of the optimisation determined sensor measurements up to 07:00, 10th January. Note the concentration along the spill path to the west and expansion up and downstream to crucial fluid measuring locations. Also observe the true and post-analysis mean paths of the spill. The optimisation determined placement has a very accurate analysis trajectory, while the ladder path placement analysis remains inaccurate. Map data ©2020 Google.

First 12 hours of sensing

With Figure 7.4 for reference: After sensors activate at 07:00 on the 10th January, error rapidly increases for pre-analysis estimates in (C) and (D) as they remove the inaccurate spill estimate and until discovery of the benchmark spill. With discovery and mapping of the benchmark spill, error decreases again and they maintain a 15% error difference until analysis, with optimisation pathing being most accurate as it focuses upon uncertain regions such as spill boundaries. The optimisation pathing is balancing measurement capability between spill mapping and fluid estimation for prediction and analysis, while the ladder pathing is solely focused upon monitoring the current-time spill. The optimisation pathing method discovers the benchmark spill as the sensor is en-route to a fluid measuring location at 09:15, while the ladder pathing discovers it at 10:15 after resorting to a large ladder-path survey across the domain. Note that for a large domain, where a large area survey may be infeasible, the ability to form an accurate analysis estimate of spill location (and thus find the true spill) is critical, and this is only achievable with sensing consideration of fluid flow.

The results of experiment (C) outperformed those of (B), with a 15% reduction in $\bar{e}_p$, indicating that the monitoring framework improves upon the existing ladder path data gathering and value replacement method. This performance gain stems from the framework improving upon the external data only prediction of future oil movement, and hence both planning a sensor route with improved coverage and maintaining better estimation of oil movement away from measurements.

Over the first 12 hours of sensing, without analysis, experiment (E) had a 20% increase in $\bar{e}_p$ over (D). The ability to only measure once every 15 minutes results in very sparse sensor measurements, though the results were still a vast improvement over a zero sensor case (A) and comparable to (B). The addition of an extra sensor improves results of (E) to be comparable to, or better (lower $\bar{e}_p$) than (D), as the 15-minute sensing frequency induced sparsity is offset by the ability to measure in two locations simultaneously.

Analysis estimation

Following 12 hours of sensing, the first analysis occurs and another forward run provides a new estimate of spill behaviour, displayed in Figure, 7.57.5.a:. For optimisation determined

sensing (D), note the reduction in $\bar{e}_p$ by up to 80% between the leak commencing and sensors activating in Figure 7.4. Also observe the removal, in analysis results, of the error spike present in no-analysis at 07:00 on the 9th of January. This spike resulted from the first measurements removing the original, poorly estimated spill, then the delay of discovery of the true spill. Now measurements confirm the presence of the analysis spill. The direction and distribution of the spill has been corrected (see Figure 7.57.5.a:) and there is overlap of spill particles with the true spill. Sensing the alternate tide cycle for 12 hours, in this mixed semi-diurnal tide location, that differs considerably from the tidal flow when the spill was released, has enabled the reduced order model to accurately estimate the fluid flow 12 hours prior to the sensor arrival.

The ladder path 12 hour analysis displays a minor 10% decrease in $\bar{e}_p$ prior to sensor activation. The ladder path sensing has solely gathered data in the west of the domain, monitoring the present-time spill, without consideration for reconstruction of the spill trajectory prior to their arrival. As a result, their analysis estimation of fluid flow around the source location is still dependent upon the external data and the analysis oil trajectory is very similar.

The next analysis occurs after 20 hours of sensing and the iterated estimate of spill behaviour is more accurate. Though only a 2% reduction in $\bar{e}_p$ for optimisation pathed sensing (D), from the 12 hour analysis, the lack of particles in the north section of the benchmark spill in Figure 7.57.5.a: is corrected. Ladder path sensing analysis (C) minorly improves, a 2% reduction in $\bar{e}_p$, but the estimation of spill particles is actually worse, closely resembling those of Figure 7.57.5.b:. The further sensing, solely concentrated to the west of the domain, has not continued the tidal flow observations of the early ladder path sensing near the spill source. Hence, the flow prediction around the source is even further based upon the incorrect external data alone. Though the analysis spill trajectory is less accurate, the prior measurements of the sensors rapidly remove the inaccurate analysis spill where it overlapped with the original pre-sensing estimate, resulting in a reduced error.

Following 24 hours of sensing the final analysis for both optimisation determined and ladder path sensors produces another estimate of the oil spill trajectory. For optimisation determined sensing (D), there is further improvement, with a 16% reduction in $\bar{e}_p$ from the 20 hour analysis, or 19% from the 12 hour analysis. The extra 4 hours of sensor data has refined the reduced order model estimate of fluid flow and the oil particles of Figure 7.57.5.b: match the benchmark spill

very closely. There is negligible spatial distribution inaccuracy between the 24 hours analysis and true spill in Figure 7.57.5.b:, and there overlap and deposition of oil on the north side of the island at 22.17N 113.8E. The remaining inaccuracy can be attributed to tidal flow differences between the measured data on the 10th of January and the tidal flow on the 9th. Note that towards the end of the first 12-hour measurement period, the 24-hour analysis has slightly higher error than the 12 hour analysis, despite having a better prediction earlier. This is simply a few erroneous particles that have yet to be corrected by measurements.

The 24 hour analysis using ladder path sensing is worse than the 20 hour analysis. The extra 4 hours of sensing has been solely to the west of the domain as sensors ladder path over the late particle positions of Figure 7.6 and measurements have sampled DMD modes far from the source origin, further over-damping the early mode amplitudes and hence the flow near the source. The western and easten flow modes have not been sampled at approximately the same time and so the DMD model does not have an accurate coupling between their amplitudes. The particles in Figure 7.57.5.b: are far from the benchmark spill and again, there is approximately no reduction in $\bar{e}_p$ from the 12 hour analysis.

The analyses of the single sensor implementation (E) outperformed those of (C) by around 15%. The sparse sensing data resulted in inaccurate particle distributions, but a close mean spill path. This indicates a good reconstruction of the main tidal amplitudes, but poor spatial accuracy in the current-flow analysis causing inaccurate dispersal of oil particles.

Sensor discussion

The sensor flight path, up to 24 hours of sensing, is displayed in Figures 7.6 and 7.7 for optimisation determined and ladder path sensing respectively. Both methods first prioritise validation of the external data prediction of a spill at 22.15N, 113.8E. The optimisation determined path, after removing the incorrectly predicted spill, goes to the source location, then to crucial fluid locations with the intention of either detecting oil or taking measurements for an accurate analysis with which to then estimate an accurate spill position. En route, the true spill is detected and the sensors map the spill by following entropy contours, then continue a monitoring strategy balancing surveillance of the spill, the spill source and critical flow regions. In contrast, the industry ladder path removes the incorrectly predicted spill and then, without other data

available, begins a ladder path survey of the entire domain in the hope of detecting a spill. This is an expensive and potentially infeasible exercise. Even with the ladder flight path covering an extra 20% in width and length of the external data provided spill, the true spill is not detected and is only mapped after discovery in a large area survey.

The optimisation determined sensor travels 37% of the distance of the industry pathing (543 to 1440 miles), with similar estimation error while sensors are active and less analysis error, demonstrating an improvement in efficiency. This is because ladder pathed sensors travel at their maximum speed to maximise the update rate of sensor measurements, while the optimisation pathed sensors move to new positions every 15 minutes and each position may or may not require flying at maximum speed.

With reference to Table 7.2, adding sensors to the ladder pathing method yields irregular results. The initial improvement is due to the increased number of measurements at spill activation providing more useful data for the Fourier estimation of tides prior to sensor arrival. However, as the number of sensors rise, the increased measurement density in the west of the domain leads to inaccurate estimation of fluid flow around the source. In contrast, each additional sensor improves the monitoring framework by at least 20%.

7.3 Source locating for an oil spill

This test scenario is identical to that of Section 7.2 unless otherwise stated. The key difference is a lack of knowledge about the spill. It is not known when or where the spill occurred. A snapshot of the spill, containing false positive and negative oil particles is captured at 05:00 on the 9th January 2020, 5 hours after the spill leaked. A sensor arrives at 07:00 on the 9th January and remains active for 24 hours. After 12 hours, 20 hours and 24 hours of sensing, the source location is estimated using a spill backtracking then forward tracking method (see Chapter 4, Section 4.3.3) and an analysis spill trajectory determined. Figures 7.9, 7.10, 7.11 and 7.12 display the source location probability distributions for pre-analysis, 12 hours, 20 hours and 24 hours of sensing, while Figures 7.13, 7.14 and 7.15 display the oil particles and flight paths. In summary, the monitoring method accurately determines the source location to within a kilometre and also accurately determines the analysis spill trajectory, while monitoring the spill with a single sensor for 24 hours. However, these results do rely upon sensors being

capable of determining oil age.

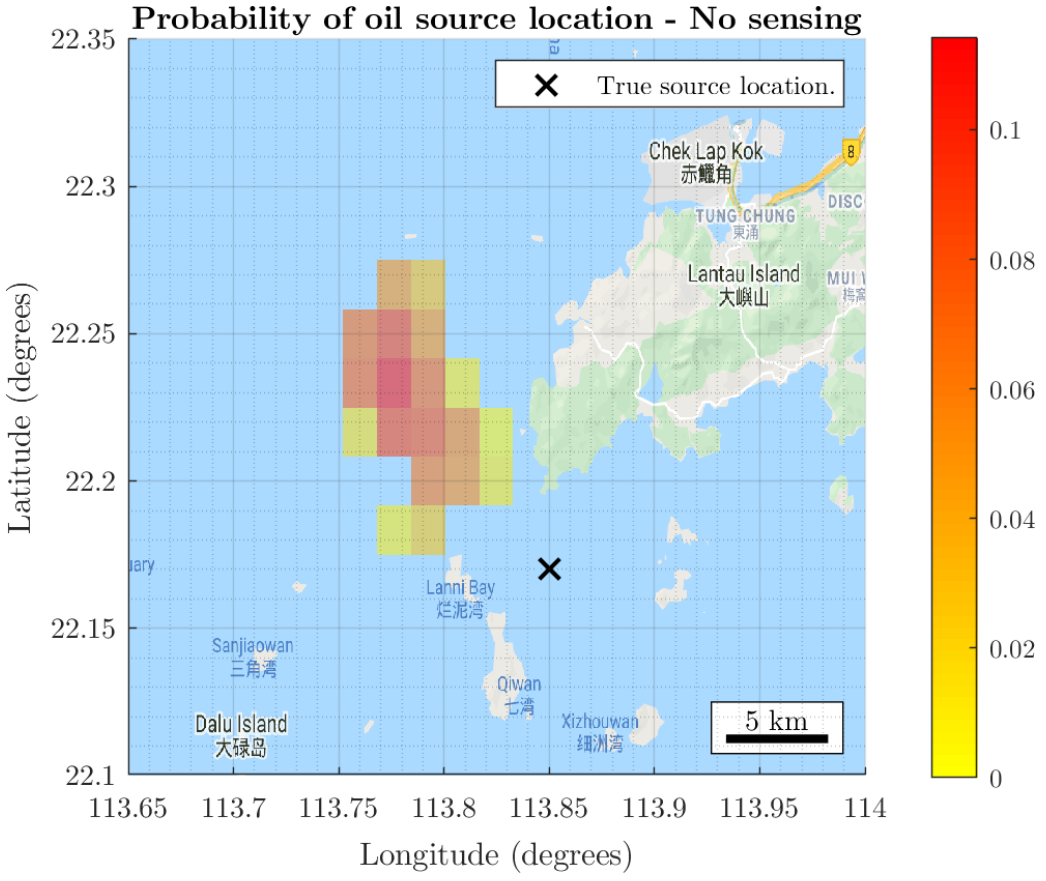


Figure 7.9: This figure displays the probability of source location across the domain, before sensing has corrected the fluid flow in the region. The inaccurate fluid flow results in a source location estimate 7 km to the north-west of the true location. Map data ©2020 Google.

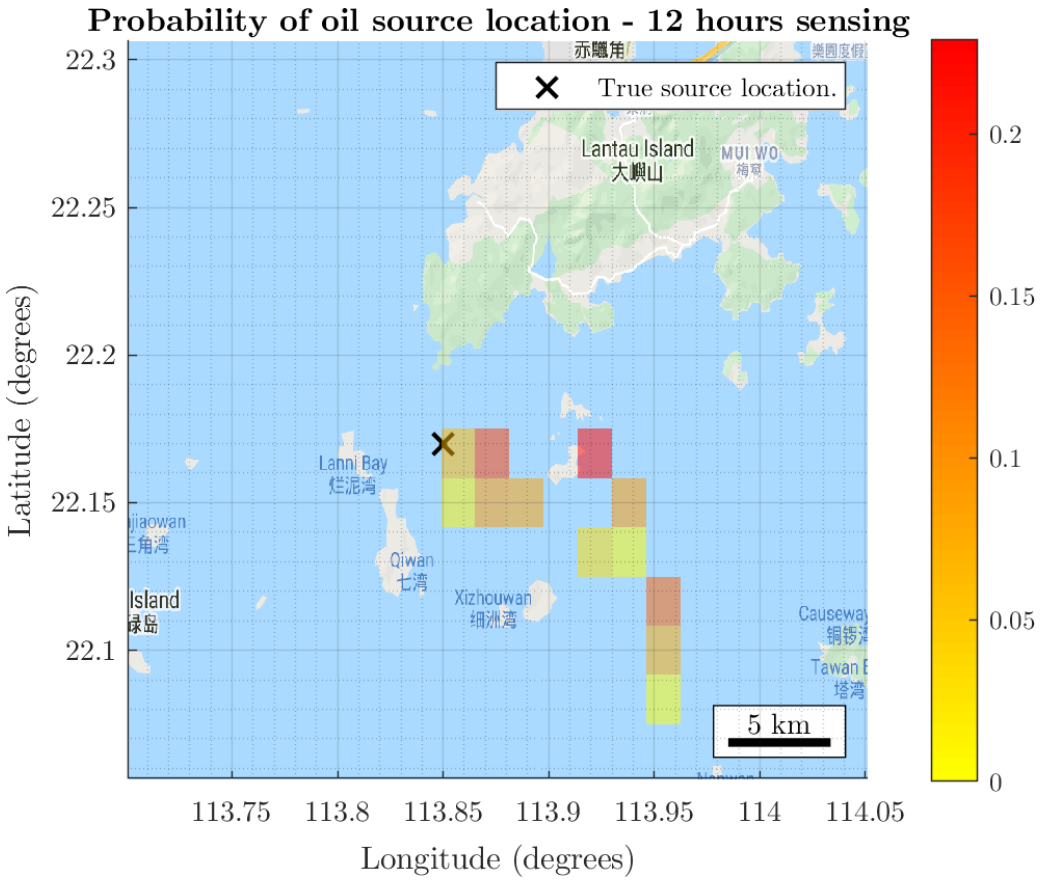


Figure 7.10: This figure displays the probability of source location across the domain, after 12 hours of sensing. The 12 hours of sensing has partially corrected the fluid flow and the source location probability distribution overlaps the true source location. The first 12 hours of sensing has focused on mapping the spill and the fluid flow down-stream, see the sensor path in Figure 7.14, but with an approximate source location identified and an analysis spill trajectory established, the next 12 hours of sensing will balance spill monitoring, down-stream flow measuring and flow measuring around the source. Map data ©2020 Google.

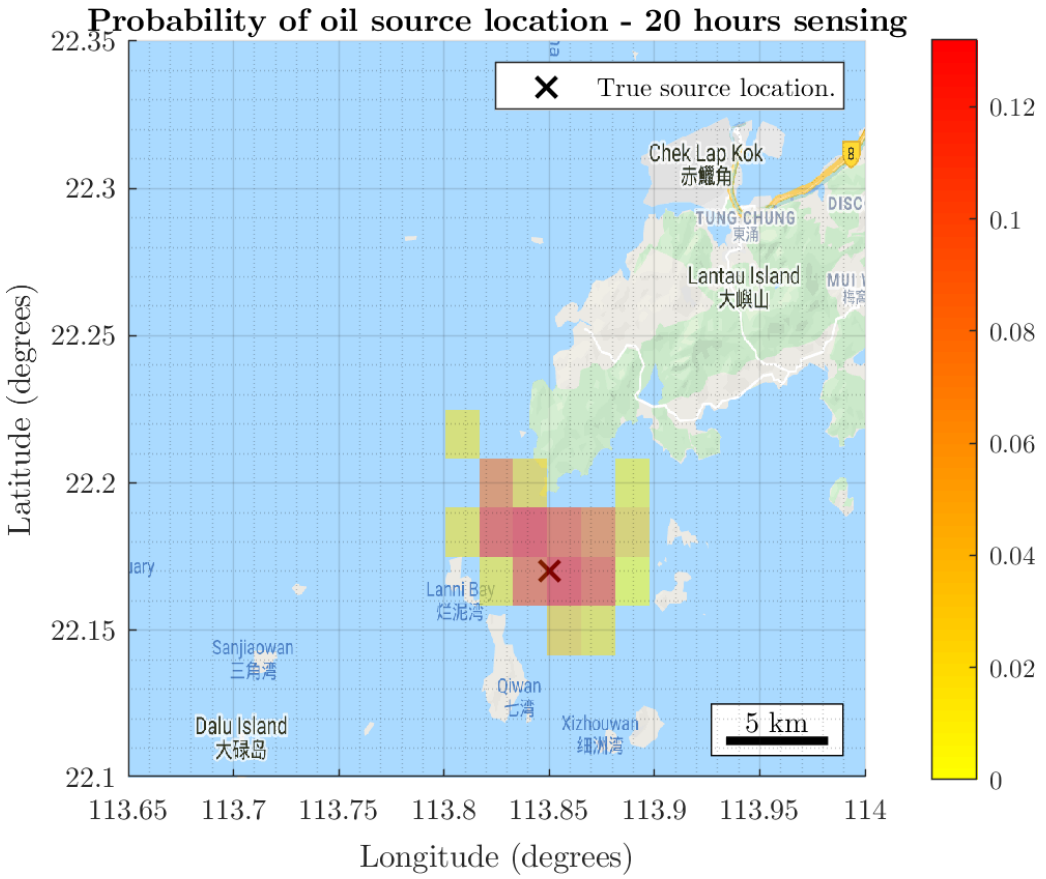


Figure 7.11: This figure displays the probability of source location across the domain, after 20 hours of sensing. A further 8 hours of sensing, with measurements around the source location, see the sensor path in Figure 7.15, has corrected the inaccurate initial fluid flow data and the resulting analysis spill and estimated source location match closely with the true benchmark spill trajectory and source. Map data ©2020 Google.

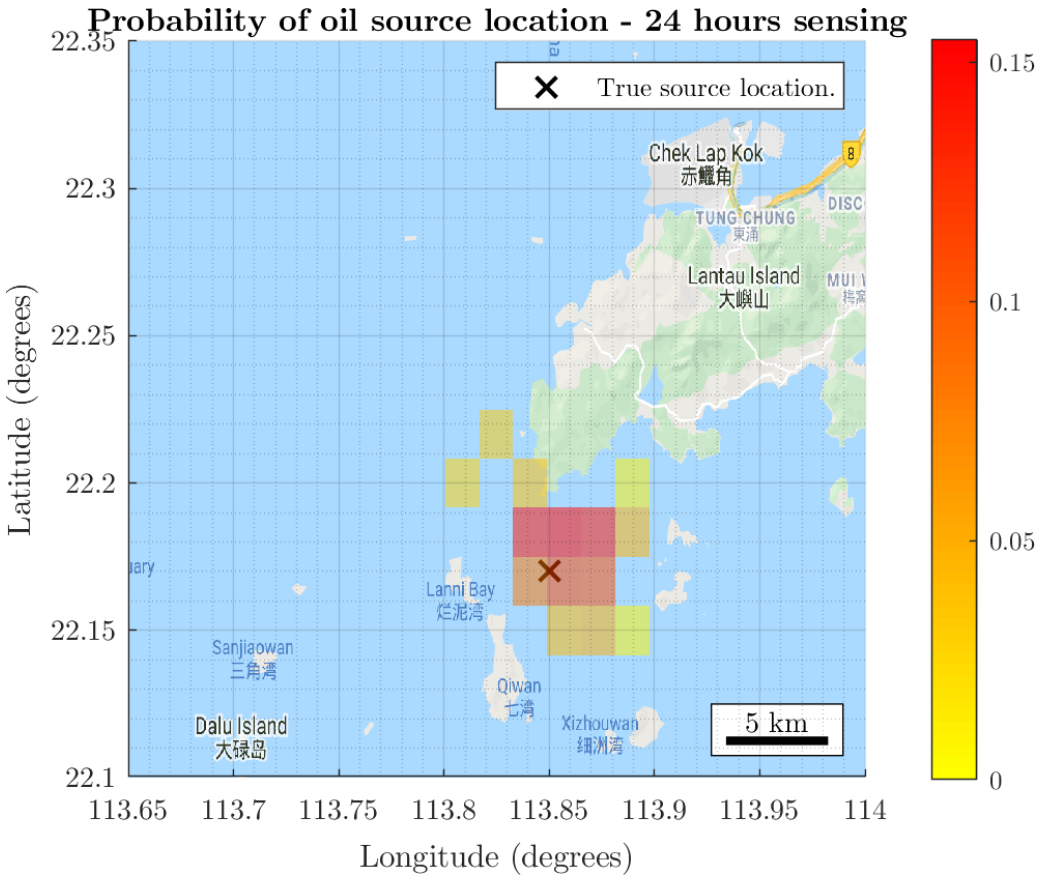


Figure 7.12: This figure displays the probability of source location across the domain, after 24 hours of sensing. The spill source probability distribution is further concentrated around the true source location and the analysis spill is now very accurate, see Figure 7.13, despite the lack of knowledge before sensor activation. Map data ©2020 Google.

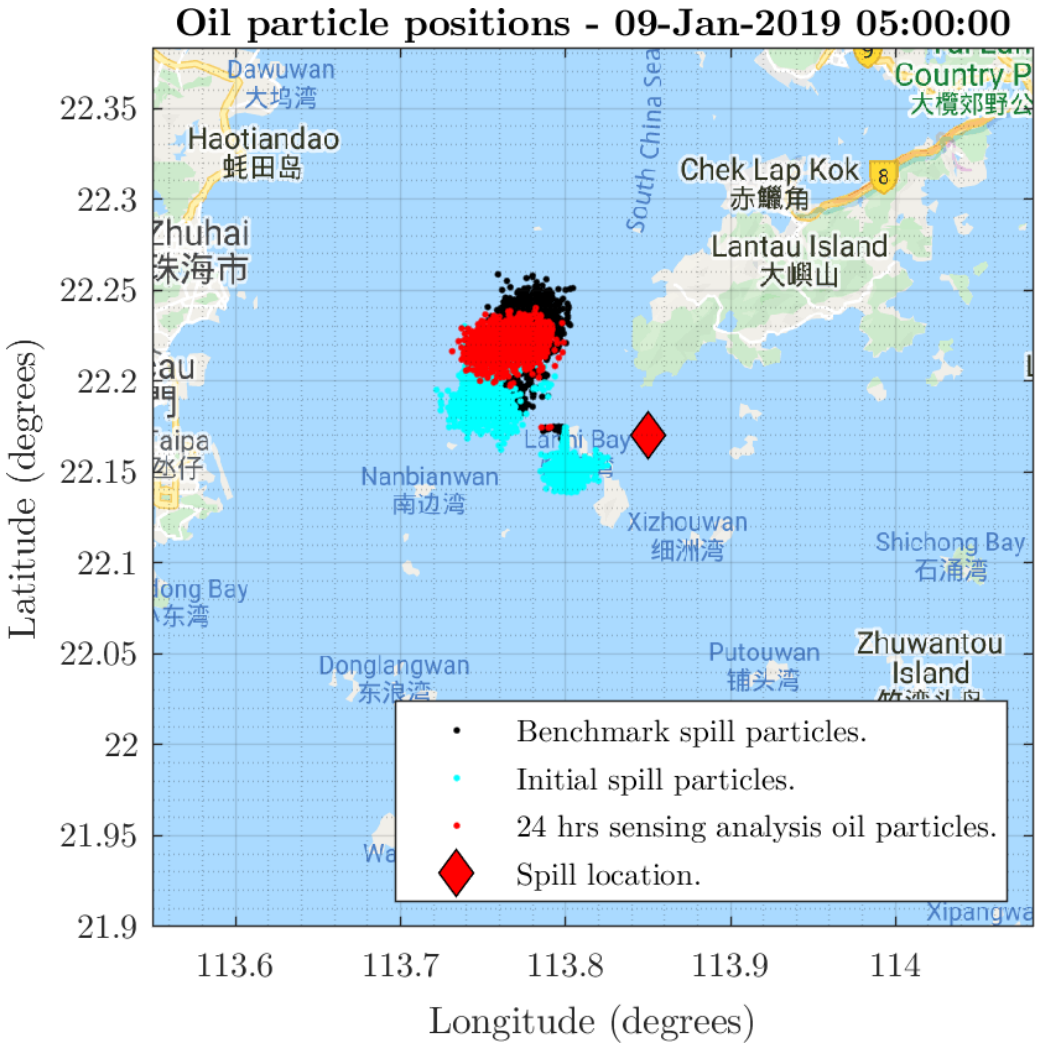


Figure 7.13: This figure displays three sets of oil particles. The black spill particles are the benchmark spill particles. The cyan particles are the initial knowledge of the spill, from a snapshot of the spill at 05:00 on the 9th January 2020, 5 hours after the spill leaked. These cyan particles contain many false positives and negatives, but have partial overlap with the true spill. The red particles are the analysis spill particles after 24 hours of sensing, which are very similarly distributed to the benchmark spill particles. Map data ©2020 Google.

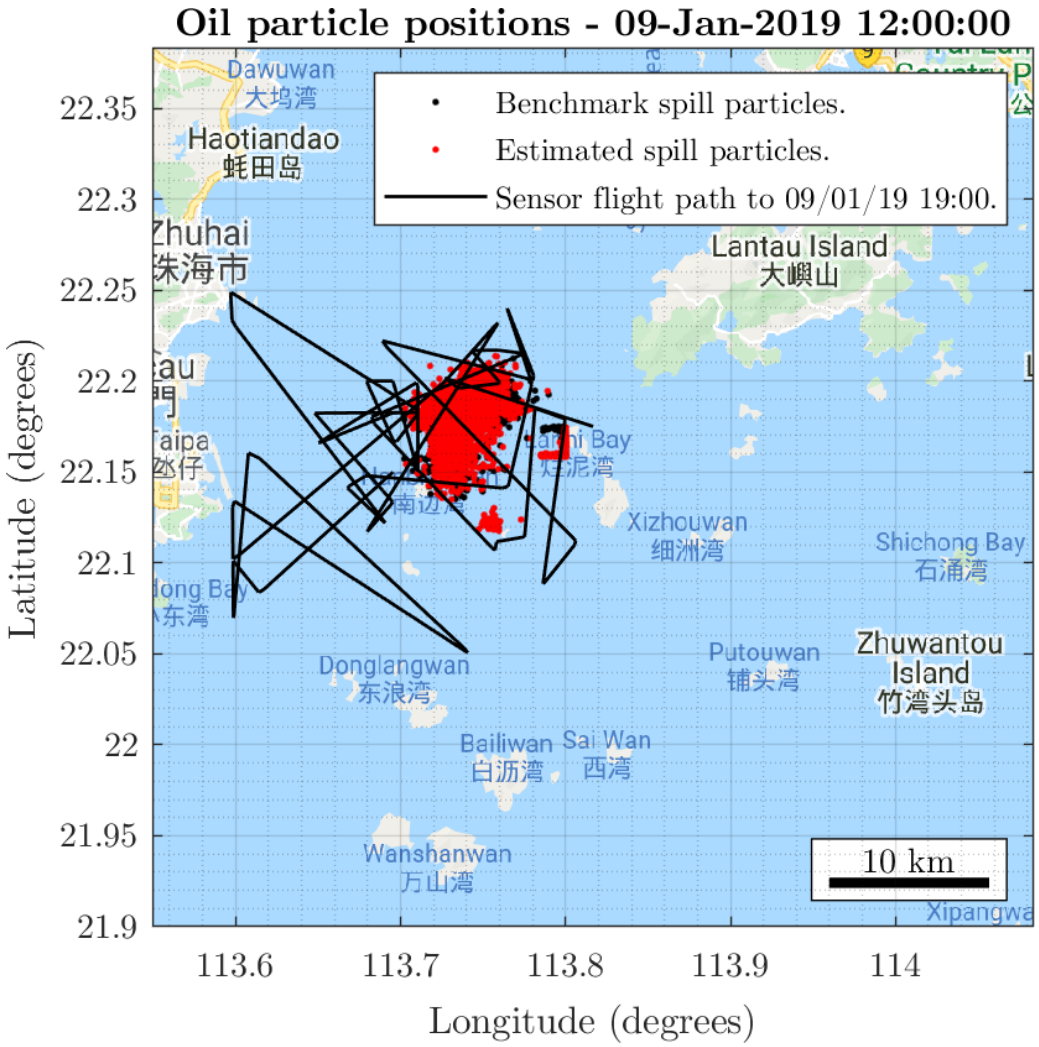


Figure 7.14: This figure displays two sets of oil particles. The black spill particles are the benchmark spill particles, the red particles are the estimated particles after 5 hours of sensing. Note the false positive cyan spill particles of Figure 7.13 have been removed and the true benchmark spill mapped. Also displayed, as a black line, is the 12 hour sensor path up to 19:00 on the 9th January. Though this sensor path extends past the particle display time in this figure, observe that the path first maps the spill, then explores south and west, continuing to monitor the spill as the particles move west, to the particle distribution in Figure 7.15. The sensor path has not explored around the source location, as an analysis source probability distribution is yet to be determined. Map data ©2020 Google.

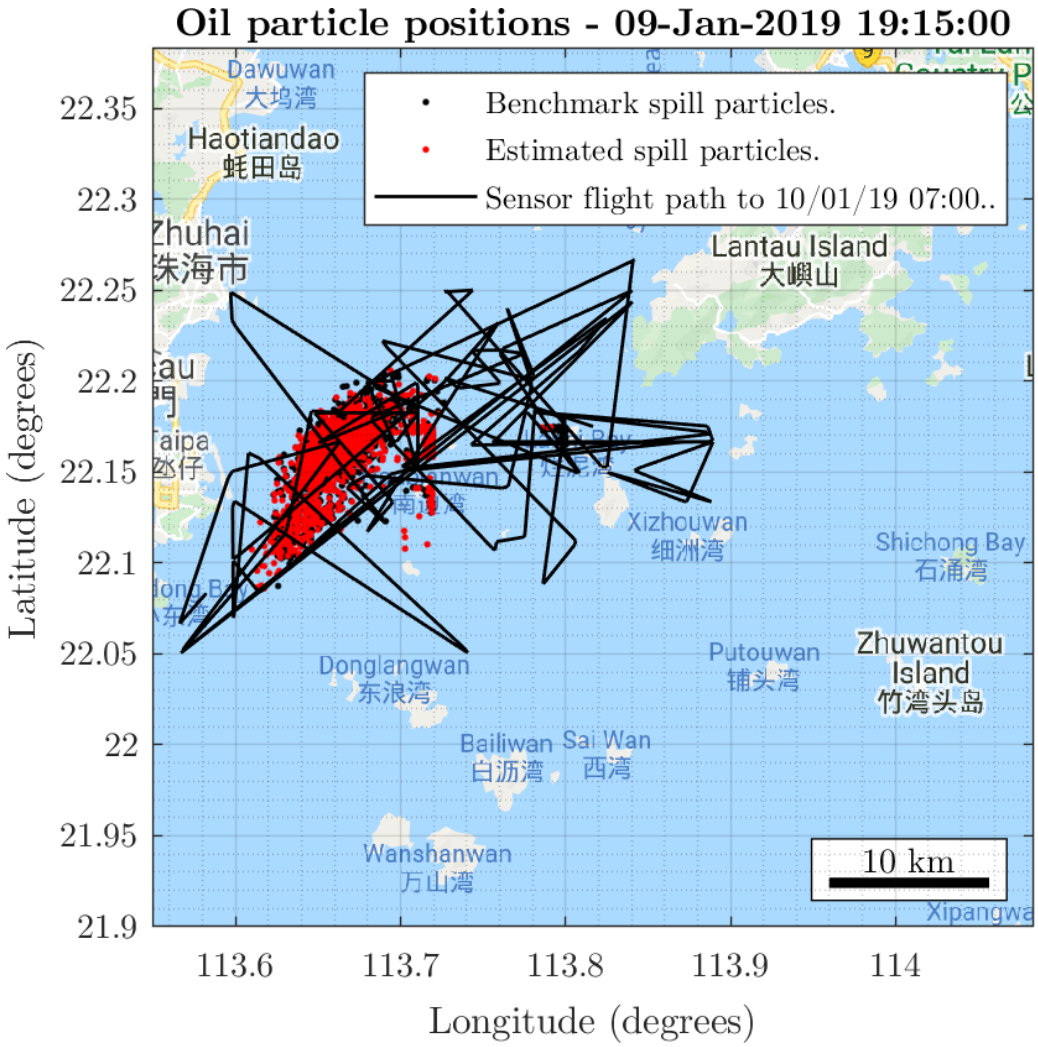


Figure 7.15: This figure displays two sets of oil particles. The black spill particles are the benchmark spill particles, the red particles are the estimated particles after 12 hours of sensing. Also displayed, as a black line, is the 24 hour sensor path up to 07:00 on the 10th January. In addition to monitoring the spill and exploring the surrounding fluid flow, the sensor also measures to the east, around the source probability distribution. Map data ©2020 Google.

Chapter 8

Conclusions and Future Work

The principal aim of this research has been the development of a contaminant monitoring framework, for gathering information with mobile sensors in the aftermath of a maritime incident, for incident mitigation and litigation. This thesis has constructed a monitoring framework utilising a bespoke environment and oil model (Chapter 4), a description of uncertainty (Chapter 4), an uncertainty minimisation for sensor placement (Chapter 5) and a data assimilation method that makes use of reduced order modelling (Chapter 6). In Chapter 7 the monitoring framework demonstrated significant improvement over the industry standard ladder-pathing method for oil spill monitoring, across monitoring and prediction, monitoring and hindcast, and monitoring and source location problem scenarios. The developed framework, though focused on oil spill monitoring, could be modified for application to other contaminants.

8.1 Main contributions

To the knowledge of the author, the key original contributions of this thesis are as follows:

1. The construction of a combined environment and oil model (named SCEM) explicitly designed to form part of a monitoring framework. It is a blend of available computationally efficient modelling techniques, from classical to modern, for the key physical processes needed to describe oil drift at sea. While ocean, wind and oil models are of course, not a new concept, SCEM is tailored to the needs of this research and is more complex than the models utilised in sensor guidance for oil spill monitoring literature. SCEM has been validated using the 2019 Grande America spill and published in Hodgson et al. 2019.

2. The modelling of the uncertainty of SCEM, a non-linear, time-varying structure system, by a further non-linear state-space system suitable for standard state-space methods. This, to the authors knowledge, is new to oil models, which usually quantify their uncertainty through an ensemble of simulations sampling from a distribution of parameters. This uncertainty also includes terms for of data-assimilation and reduced order modelling error, with further terms for describing sensor dynamics within one time step, enabling one-step-ahead optimisation.

3. Determining the optimal placement of sensors to monitor the spill, through the forming and solving of an uncertainty minimisation optimisation. Though uncertainty minimisation exists in sensor-placement literature, the cost-function terms for sensor constraints and weighting of uncertainty minimisation by information theoretic measures have been formulated for the specific requirements of oil monitoring. Unlike prior mobile sensor guidance for oil spill monitoring, the strategy here strikes a balance between oil observation and measuring the environment to improve the hind-casting or prediction of the spill using SCEM.

4. Extension of the adjoint method to incorporate bounded states, implemented as extra matrices as opposed to modification of the underlying state-space model. This provided gradient information to an optimisation solver based on prior work.

5. Use of parallel optimisations across multiple time-step sizes as an alternative to adjoint-MPC. Parallel optimisation has no guarantee of finding a global optimal, but is less memory intensive.

6. Improving an ensemble and tangent-linear model free data assimilation method, in-terms of reduced order model creation and accuracy, by employing a weighted decomposition and a “data-driven model versus trajectory” based description of error.

7. Combining all of the above into a monitoring framework, with a suitable structure for a variety of scenarios.

8. Application of the monitoring framework to example monitoring problems in high fidelity simulations. The monitoring framework demonstrated significant improvement, in error

reduction and sensor distance travelled reduction, compared to the industry standard survey method, as displayed in conference poster sessions at Intcatch 2019 and AGU Fall Meeting 2019, published in Hodgson, Esnaola, and Jones 2020b and submitted in Hodgson, Esnaola, and Jones 2020a.

8.2 Future work

The below list states possible future improvements and avenues of further research based on this work:

1. SCEM has been used to simulate the 2018 Sanchi spill in the China sea, the 2019 Aulac Fortune spill near Hong Kong and the 2020 unknown-origin spill near Brasil with the industry partner. Further model development will adapt the model for prediction of drifting bodies and other contaminants, with accompanying alteration of the uncertainty description. In August 2020, SCEM was utilised to produce a probability of location map for a man-overboard situation near Hong Kong.

2. Future work should compare the one-step-ahead control method with the adjoint-MPC method. Alternative optimisation or control methods that provide a guarantee of optimality should also be investigated. Approximating Sequence of Riccati Equation (ASRE) control was implemented, but failed to converge to a solution, likely due to the high state-dimension and degree of non-linearity present. This deserves further analysis.

3. Extension of the uncertainty description to 3-dimensions, where-by the uncertainty area becomes an uncertainty volume to be minimised. This would then be evaluated as guidance for sub-surface sensing.

4. In this work, a regularly sized staggered grid spatial discretisation was used, as it offered flexibility with meshing and boundary conditions, and was accompanied by Euler or Runge-Kutta based time discretisation (all finite difference based methods). It could perhaps offer computation speed benefits to make use of spectral methods, e.g, Galerkin projection, but this may complicate boundary conditions.

5. Perhaps most importantly, the framework needs to be utilised in industry, with suitable

hardware and sensors. This is a major undertaking, requiring considerable resources, but would deliver tangible impact.

Bibliography

Akbarzadeh, Vahab et al. (2014). "Efficient sensor placement optimization using gradient descent and probabilistic coverage". In: *Sensors (Switzerland)* 14.8, pp. 15525–15552.

Al-Rabeh, A. H., H. M. Cekirge, and N. Gunay (1989). "A stochastic simulation model of oil spill fate and transport". In: *Applied Mathematical Modelling* 13.6, pp. 322–329.

Alejo, D. et al. (2015). "Collision-free trajectory planning based on Maneuver Selection-Particle Swarm Optimization". In: *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 72–81.

Allianz (2015). *Safety and Shipping Review 2015*. Tech. rep. January 2015, p. 36.

Amezcua, Javier, Michael Goodliff, and Peter Jan Van Leeuwen (2017). "A weak-constraint 4DEnsembleVar. Part I: Formulation and simple model experiments". In: *Tellus, Series A: Dynamic Meteorology and Oceanography* 69.1.

Annoni, J et al. (2018). "Sparse-Sensor Placement for Wind Farm Control Sparse-Sensor Placement for Wind Farm Control". In:

Armijo, Larry (1966). "Minimisation Of Functions Having Lipschitz Continuous First Partial Derivatives". In: *Pacific Journal Of Mathematics* 16.1, pp. 1–3.

Bae Ji-sook (2007). *Clearing Oil Contamination to Take at Least Two Months*.

Baldauf, M and G Zängl (2012). "Horizontal nonlinear Smagorinsky diffusion". In: *COSMO newsletter* 2.April, pp. 3–7.

Banerjee, Abhijit, Dipendranath Ghosh, and Suvrojit Das (2018). "Modified firefly algorithm for area estimation and tracking of fast expanding oil spills". In: *Applied Soft Computing Journal* 73, pp. 829–847.

Batchelder, Harold P. (2006). "Forward-in-time-/backward-in-time-trajectory (FITT/BITT) modeling of particles and organisms in the coastal ocean". In: *Journal of Atmospheric and Oceanic Technology* 23.5, pp. 727–741.

Bertuccelli, L. F. and J. P. How (2005). "Robust UAV search for environments with imprecise probability maps". In: *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference, CDC-ECC '05* 2005, pp. 5680–5685.

Bertuccelli, Luca F et al. (2009). "Real-time Multi-UAV Task Assignment in Dynamic and Uncertain Environments". In: *Construction* August, pp. 1–16.

Boufadel, Michel C. et al. (2007). "Lagrangian simulation of oil droplets transport due to regular waves". In: *Environmental Modelling and Software* 22.7, pp. 978–986.

Braley, Colin and Adrian Sandu (2009). *Fluid Simulation For Computer Graphics : A Tutorial in Grid Based and Particle Based Methods*.

Breivik, Øyvind et al. (2012). "BAKTRAK: Backtracking drifting objects using an iterative algorithm with a forward trajectory model". In: *Ocean Dynamics* 62.2, pp. 239–252.

Bruemmer, David J. et al. (2002). "A Robotic Swarm for Spill Finding and Perimeter Formation". In: *Spectrum* Reno, NV.Aug. arXiv: 1011.1669v3.

Brunton, Steven L. et al. (2015). "Compressed Sensing And Dynamic Mode Decomposition". In: *Journal of Computational Dynamics* 2.2, pp. 165–191.

Candido, Salvatore and Seth Hutchinson (2011). "Minimum uncertainty robot navigation using information-guided POMDP planning". In: *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 6102–6108.

Carlowicz, Mike (2010). *NASA Deploys Planes, Targets Satellites to Aid in Oil Spill Response*.

CERC (2017). *URBAN CANOPY FLOW SPECIFICATION*. Tech. rep. August, pp. 1–7.

Cessna, Joseph, Student Member, and Thomas Bewley (2010). "A hybrid (variational/Kalman) ensemble smoother for the estimation of nonlinear high-dimensional discretizations of PDE systems". In: pp. 1–15.

Cevik, Polat et al. (2013). "The Small and Silent Force Multiplier : A Swarm UAV — Electronic Attack". In: pp. 595–608.

Chao, Xiaobo, N. Jothi Shankar, and Sam S. Y. Wang (2003). "Development and Application of Oil Spill Model for Singapore Coastal Waters". In: *Journal of Hydraulic Engineering* 129.7, pp. 495–503.

Chen, Changsheng, Hedong Liu, and Robert C. Beardsley (2003). "An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: Application to coastal ocean and estuaries". In: *Journal of Atmospheric and Oceanic Technology* 20.1, pp. 159–186.

Chin, T. M., A. C. Haza, and A. J. Mariano (2002). "A reduced-order information filter for multilayer shallow-water models: Profilling and assimilation of sea surface height". In: *Journal of Atmospheric and Oceanic Technology* 19.4, pp. 517–533.

Chong, Edwin K P, Chris Kreucher, and Alfred Hero (2007). "POMDP APPROXIMATION USING SIMULATION AND HEURISTICS". In: *Foundations and Applications of Sensor Management*. October, pp. 93–117.

Ciappa Achille; Costabile, Salvatore (2014). "Oil spill hazard assessment using a reverse trajectory method for the Egadi marine protected area (Central Mediterranean Sea)". In: *Marine Pollution Bulletin* 84.1-2, pp. 44–55.

Cindori, Mihael et al. (2020). "The Atmospheric Boundary Layer Above Generic Hills: Computational Model of a Unidirectional Body Force-Driven Flow". In: *Boundary-Layer Meteorology* 176.2, pp. 159–196.

Clark, Emily, J. Nathan Kutz, and Steven L. Brunton (2020). "Sensor Selection With Cost Constraints for Dynamically Relevant Bases". In: *IEEE Sensors Journal* 20.19, pp. 11674–11687.

Coope, Ian and Christopher Price (1995). "A modified BFGS formula maintaining positive definiteness with Armijo-Goldstein steplengths". In: *Journal of Computational Mathematics* 13.2, pp. 156–160.

Courant, Richard, Kurt Friedrichs, and Hans Lewy (1967). "On the partial difference equations of mathematical physics". In: *IBM journal of Research and Development* 11.2, pp. 215–234.

Daescu D. N., Navon, I. M. (2006). "Efficiency of a POD-based reduced second-order adjoint model in 4D-Var data assimilation D." In: *International Journal for Numerical Methods in Fluids*.

Dasa, S and A Neumaiera (2011). *Fast Regularized Low Rank Approximation of Weighted Data Sets*.

De Dominicis, M. et al. (2013a). "MEDSLIK-II, a Lagrangian marine surface oil spill model for short-term forecasting-Part 1: Theory". In: *Geoscientific Model Development* 6.6, pp. 1851–1869.

De Dominicis, M. et al. (2013b). "MEDSLIK-II, a Lagrangian marine surface oil spill model for short-term forecasting-Part 2: Numerical simulations and validations". In: *Geoscientific Model Development* 6.6, pp. 1871–1888.

Delvigne, G. A L and C. E. Sweeney (1988). "Natural dispersion of oil". In: *Oil and Chemical Pollution* 4.4, pp. 281–310.

Dijkstra, E. W. (1959). "A note on two problems in connexion with graphs". In: *Numerische Mathematik* 1.1, pp. 269–271.

Duan, Hai bin et al. (2009). "Max-Min Adaptive Ant Colony Optimization Approach to Multi-UAVs Coordinated Trajectory Replanning in Dynamic and Uncertain Environments". In: *Journal of Bionic Engineering* 6.2, pp. 161–173.

Duran, Rodrigo et al. (2018). "Simulation of the 2003 Foss Barge - Point wells oil spill: A comparison between BLOSOM and GNOME oil spill models". In: *Journal of Marine Science and Engineering* 6.3.

E-geos (2018). *The Sanichi's incident in the Chinese Sea: a case study.*

Eaton, Christopher, Edwin Chong, and Anthony Maciejewski (2016). "Multiple-Scenario Unmanned Aerial System Control: A Systems Engineering Approach and Review of Existing Control Methods". In: *Aerospace* 3.1, p. 1.

Ehsan, Ali and Qiang Yang (2019). "State-of-the-art techniques for modelling of uncertainties in active distribution network planning: A review". In: *Applied Energy* 239.November 2018, pp. 1509–1523.

Elliott, Alan J. (1986). "Shear diffusion and the spread of oil in the surface layers of the North Sea". In: *Deutsche Hydrographische Zeitschrift* 39.3, pp. 113–137.

F. Harlow and J. Welch (1965). "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid With Free Surface". In: *Physics of Fluids* 8.1, pp. 2182–2189.

Fay, J.A. (1971). "Physical Processes in the Spread of Oil on a Water Surface". In: *International Oil Spill Conference Proceedings* 1971.1, pp. 463–467.

Feder, Hans Jacob S., John J. Leonard, and Christopher M. Smith (1999). "Adaptive mobile robot navigation and mapping". In: *International Journal of Robotics Research* 18.7, pp. 650–668.

Filho, Enio Vasconcelos and Paulo Lopes (2019). "A Dynamic Mode Decomposition Approach With Hankel Blocks to Forecast Multi-Channel Temporal Series". In: *IEEE Control Systems Letters* 3.3, pp. 739–744.

Fingas, Merv (2015). "Oil and Petroleum Evaporation". In: *Handbook of Oil Spill Science and Technology* 2.3, pp. 205–223. eprint: 9809069v1 (gr-qc).

Fingas, Merv and Carl Brown (2014). "Review of oil spill remote sensing". In: *Marine Pollution Bulletin* 83.1, pp. 9–23.

Fingas, Merv and Carl E. Brown (2018). "A review of oil spill remote sensing". In: *Sensors (Switzerland)* 18.1, pp. 1–18.

Fingas, Merv F. (2012). "Studies on the Evaporation Regulation Mechanisms of Crude Oil and Petroleum Products". In: *Advances in Chemical Engineering and Science* 02.April, pp. 246–256.

Fingas, Mervin and Carl Brown (1998). "Review of Oil Spill Remote Sensing". In: *Spill Science & Technology Bulleti* 4.4, pp. 199–208.

Fiorentino, C and M Virelli (2016). *COSMO-SkyMed Mission and Products Description.*

Foxwell, David (2017). *Oil spill responders expect growing role for unmanned aerial vehicles.*

French McCay, D. et al. (2016). "Modeling Oil Fate and Subsurface Exposure Concentrations from the Deepwater Horizon Oil Spill". In: *AMOP Technical Seminar.*

Funke, S. W., P. E. Farrell, and M. D. Piggott (2014). "Tidal turbine array optimisation using the adjoint approach". In: *Renewable Energy* 63, pp. 658–673. arXiv: 1304.1768v1.

Galt, J A (1984). "A Finite-Element Solution Procedure for the Interpolation of Current Data in Complex Regions". In: *Journal of Physical Oceanography* 10.

Galt, J A and D L Payton (2005). "The use of receptor mode trajectory analysis techniques for contingency planning". In: *2005 International Oil Spill Conference, IOSC 2005*, p. 8864.

Galt, J.A. A and Roy Overstreet (2011). *Development of Spreading Algorithms for the ROC.* Tech. rep.

Geng, L. et al. (2013). "Cooperative task planning for multiple autonomous UAVs with graph representation and genetic algorithm". In: *10th IEEE International Conference on Control and Automation, ICCA*, pp. 394–399.

Glover, Fred (1990). "Tabu Search: A Tutorial". In: *Interfaces* 1990.August 1990, pp. 74–94.

Golding, Bw et al. (2001). "EUMETSAT position paper on observation requirements for now-casting and very short range forecasting in 2015-2025". In: *WORLD METEOROLOGICAL ORGANIZATION*.

Goldstein, A A (1965). "On Steepest Descent". In: *SIAM Journal on Control* 3, pp. 147–151.

Gómez, Cristina and David R Green (2017). "Small unmanned airborne systems to support oil and gas pipeline monitoring and mapping". In: *Arabian Journal of Geosciences*.

Goncalves, Rafael C. et al. (2016). "A framework to quantify uncertainty in simulations of oil transport in the ocean." In: *Journal of Geophysical Research: Oceans*, pp. 2058–2077.

Grubesic, Tony H., Ran Wei, and Jake Nelson (2017). "Optimizing oil spill cleanup efforts: A tactical approach and evaluation framework". In: *Marine Pollution Bulletin* 125.1-2, pp. 318–329.

Heins, Peter H and Bryn Ll Jones (2016). "SWEM : A Multiphysics Sea-Surface Simulation Environment". In: *UKACC 11th International Conference on Control*.

Hinson, B.T. (2014). "Observability-based guidance and sensor placement". In: *PhD Thesis, Aeronautics and Astronautics, University of Washington*, pp. 1–147.

Hirsh, Seth M et al. (2020). "Centering data improves the dynamic mode decomposition". In: *SIAM Journal on Applied Dynamical Systems* 19.3, pp. 1920–1955.

Hodgson, Zak, Inaki Esnaola, and Bryn Jones (2019). "Optimal Model-Based Sensor Placement & Adaptive Monitoring Of An Oil Spill". In: arXiv: `1911.10823`.

Hodgson, Zak, Iñaki Esnaola, and Bryn Jones (2020a). "Model-Based Optimal Adaptive Monitoring of Oil". In: *IEEE Transactions on Control Systems Technology - Submitted*.

— (2020b). "Optimal Model-Based Sensor Placement & Adaptive Monitoring Of An Oil Spill". In: *IFAC-V*.

Hodgson, Zak et al. (2019). "A combined ocean and oil model for model-based adaptive monitoring". In: arXiv: `1910.12658`.

Hollinger, Geoffrey A. and Gaurav S. Sukhatme (2014). "Sampling-based robotic information gathering algorithms". In: *International Journal of Robotics Research* 33.9, pp. 1271–1287.

Hoult, David P (1972). "Oil spreading on the sea". In: *Massachusetts Institute of Technology* 4.1, pp. 341–368.

Hover, Franz and Harrison Chin (2009). *2.017J Design of Electromechanical Robotic Systems, Fall*. Tech. rep. Massachusetts Institute of Technology: MIT OpenCourseWare, https://ocw.mit.edu/. License: Creative Commons BY-NC-SA.

Hu, Jinwen et al. (2013). "Multiagent information fusion and cooperative control in target search". In: *IEEE Transactions on Control Systems Technology* 21.4, pp. 1223–1235.

Hu, Yuanming et al. (2019). "DiffTaichi: Differentiable Programming for Physical Simulation". In: *ICLR 2020*, pp. 1–20. arXiv: 1910.00935.

Hua, Dai and Peter Lancaster (1996). "Linear Matrix Equations From an Inverse Problem of Vibration Theory". In: *Linear Algebra and its Applications* 246.1, pp. 31–42.

Hunter, J. R., P. D. Craig, and H. E. Phillips (1993). "On the use of random walk models with spatially variable diffusivity". In: *Journal of Computational Physics* 106.2, pp. 366–376.

Hutchinson, Michael, Cunjia Liu, and Wen Hua Chen (2019). "Information-Based Search for an Atmospheric Release Using a Mobile Robot: Algorithm and Experiments". In: *IEEE Transactions on Control Systems Technology* 27.6, pp. 2388–2402.

Ichiye, Takashi (1967). "Upper ocean boundary-layer flow determined by dye diffusion". In: *Physics of Fluids* 10.9.

IPIECA and IOGP (2016). "Aerial observation of oil spills at sea". In: *IPIECA resources*.

Isobe, Atsuhiko et al. (2009). "Two-way particle-tracking model for specifying sources of drifting objects: Application to the East China Sea shelf". In: *Journal of Atmospheric and Oceanic Technology* 26.8, pp. 1672–1682.

ITOPF (2011a). *Aerial Observation of Marine Oil Spills*. Tech. rep. 1. ITOPF, p. 12.

— (2011b). "TIP 01: Aerial Observation of Marine Oil Spills". In: *ITOPF Technical Information Paper* 1, p. 12.

— (2011c). "TIP 02: Fate of Marine Oil Spills". In: *ITOPF Technical Information Paper*.

— (2014). *Aerial Surveillance - ITOPF*.

— (2017). “Oil tanker spill statistics 2016”. In: *The International Tanker Owners Pollution Federation Limited* February.

— (2019). *Oil Tanker Spill Statistics 2018*. Tech. rep. January. ITOPF, p. 12.

Iungo, G V, M Abkar, and F Port (2015). “Data-driven Reduced Order Model for prediction of wind turbine wakes”. In: 625, pp. 1–11.

Jayaraman, Balaji and S. M.Abdullah Al Mamun (2020). “On data-driven sparse sensing and linear estimation of fluid flows”. In: *Sensors (Switzerland)* 20.13, pp. 1–31.

Jha, Maya Nand, Jason Levy, and Yang Gao (2008). “Advances in Remote Sensing for Oil Spill Disaster Management: State-of-the-Art Sensors Technology for Oil Spill Surveillance”. In: *Sensors* 8.1, pp. 236–255.

Joshi, Siddharth and Stephen Boyd (2009). “Sensor selection via convex optimization”. In: *IEEE Transactions on Signal Processing* 57.2, pp. 451–462.

Juszczak, P, D M J Tax, and Robert P W Duin (2000). *Feature scaling in support vector data description*. Tech. rep.

Kakalis, Nikolaos M P and Yiannis Ventikos (2008). “Robotic swarm concept for efficient oil spill confrontation”. In: 154, pp. 880–887.

Kaymal, Turgut (2016). “Unmanned Aircraft Systems for Maritime Operations”. In: *International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 763–768.

Khang, Nguyen Van (2012). “Partial Derivative of Matrix Functions With Respect To a Vector Variable”. In: *Vietnam Journal of Mechanics* 30.4, pp. 269–279.

Kim, Tae Ho et al. (2014). “Analysis of the contribution of wind drift factor to oil slick movement under strong tidal condition: Hebei Spirit oil spill case”. In: *PLoS ONE* 9.1, pp. 1–14.

Kirincich, Anthony R. and John A. Barth (2008). “Time-Varying Across-Shelf Ekman Transport and Vertical Eddy Viscosity on the Inner Shelf”. In: *Journal of Physical Oceanography* 39.3, pp. 602–620.

Kitano, H et al. (1999). “RoboCup Rescue: search and rescue in large-scale disasters as a domain for autonomous agents research”. In: *IEEE SMC’99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No.99CH37028)*.

Koopman, B. O. (1931). “Hamiltonian Systems and Transformation in Hilbert Space”. In: *Proceedings of the National Academy of Sciences* 17.5, pp. 315–318.

Krishnamoorthy, K., D. Casbeer, and M. Pachter (2015). "Minimum time UAV pursuit of a moving ground target using partial information". In: *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 204–208.

Krishnamoorthy, K. et al. (2012). "UAV search & capture of a moving ground target under delayed information". In: *Proceedings of the IEEE Conference on Decision and Control*, pp. 3092–3097.

Kuwata, Yoshiaki and Jonathan P. How (2007). "Robust cooperative decentralized trajectory optimization using receding horizon MILP". In: *Proceedings of the American Control Conference* 4, pp. 522–527.

Lamb, Horace (1895). "Hydrodynamics". In: *Journal of Chemical Information and Modeling* 53.9, pp. 468–470. arXiv: `1011.1669v3`.

Lardner, R W and N Gunay (2000). "Gulfspill Version 2 . 0 : a software package for oil spills in the Arabian Gulf". In: *Environment Modelling & Software* 15, pp. 425–442.

Laruelle, Franck (2009). "The Role of ITOPF & Hebei Spirit case study". In: *Oil Clean Conference*.

— (2011). "Responding to Spills in Remote Locations: GULSER ANA (Madagascar) & OLIVA (South Atlantic)". In: *ITOPF Library*, pp. 1–12.

Lehr, W. et al. (2002). "Revisions of the ADIOS oil spill model". In: *Environmental Modelling & Software* 17.2, pp. 189–197.

Lehr, W. J. et al. (1984). "A new technique to estimate initial spill size using a modified fay-type spreading formula". In: *Marine Pollution Bulletin* 15.9, pp. 326–329.

Leroux, Romain and Laurent Cordier (2016). "Dynamic mode decomposition for non-uniformly sampled data". In: *Experiments in Fluids* 57.5.

Leyder, Claude et al. (2018). "Optimal sensor placement methods and metrics–comparison and implementation on a timber frame structure". In: *Structure and Infrastructure Engineering* 14.7, pp. 997–1010.

Li, Yan, Jiang Zhu, and Hui Wang (2013). "The impact of different vertical diffusion schemes in a three-dimensional oil spill model in the Bohai Sea". In: *Advances in Atmospheric Sciences* 30.6, pp. 1569–1586.

Lian, Wu et al. (2018). "An improved adaptive sampling algorithm". In: *2018 IEEE 4th International Conference on Computer and Communications, ICCC 2018*, pp. 2205–2211.

Liu, Kang, Ren Jun Yan, and C. Guedes Soares (2018). "Optimal sensor placement and assessment for modal identification". In: *Ocean Engineering* 165.July, pp. 209–220.

Lonin, Serguei A. (1999). "Lagrangian model for oil spill diffusion at Sea". In: *Spill Science and Technology Bulletin* 5.5-6, pp. 331–336.

Mackay, Donald and Ronald S. Matsugu (1973). "Evaporation rates of liquid hydrocarbon spills on land and water". In: *The Canadian Journal of Chemical Engineering* 51.4, pp. 434–439.

Madec, Gurvan (2011). *NEMO Ocean Engine: version 3.3*. Tech. rep. l'Institut Pierre-Simon Laplace.

Marsh, Charles (2013). "Introduction to Continuous Entropy". In: *Department of Computer Science Princeton University crmarsh@princeton.edu*, pp. 1–17.

Marsooli, Reza (2017). "Some questions on "A Coupled Circulation-Wave Model for Numerical Simulation of Storm Tides and Waves (2017)"". In: *[email]*.

Marsooli, Reza et al. (2017). "A Coupled Circulation-Wave Model for Numerical Simulation of Storm Tides and Waves". In: *Journal of Atmospheric and Oceanic Technology*.

Mattingley, Jacob, Yang Wang, and S P Boyd (2011). "Receding Horizon Control". In: *IEEE Control Systems* 31.3, pp. 52–65. arXiv: `1011.1669v3`.

Mellor, George (2003). "The Three-Dimensional Current and Surface Wave Equations". In: *Journal of Physical Oceanography* 33.9, pp. 1978–1989.

Mellor, George L., Mark a. Donelan, and Lie-Yauw Oey (2008). "A Surface Wave Model for Coupling with Numerical Ocean Circulation Models". In: *Journal of Atmospheric and Oceanic Technology* 25.10, pp. 1785–1807.

Meng, Bo Bo, Xiaoguang Gao, and Yunhui Wang (2009). "Multi-mission path re-planning for multiple unmanned aerial vehicles based on unexpected events". In: *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2009* 1, pp. 423–426.

Michalska, Hannah and David Q Mayne (1995). "Moving Horizon Observers and Observer-Based Control". In: *IEEE Transactions on Automatic Control* 40.6.

Montgomery, D. C. and G. C. Runger (1994). *Applied statistics and probability for engineers*. John Wiley, p. 201.

Moon, Sangwoo et al. (2015). "Decentralized information-theoretic task assignment for searching and tracking of moving targets". In: *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 1031–1036.

Moreland, Erin E et al. (2015). "Evaluation of a ship-based unoccupied aircraft system (UAS) for surveys of spotted and ribbon seals in the Bering Sea pack ice 1". In: *J. Unmanned Veh. Sys.* 3.3, pp. 114–122.

Morlier, Joseph et al. (2018). "An EGO-like optimization framework for sensor placement optimization in modal analysis". In: *Smart Materials and Structures* 27.7.

Nelson, Jake R. and Tony H. Grubesic (2019). "Oil spill modeling: computational tools, analytical frameworks, and emerging technologies". In: *Progress in Physical Geography* 43.1, pp. 129–143.

— (2020). "Oil spill modeling: Mapping the knowledge domain". In: *Progress in Physical Geography* 44.1, pp. 120–136.

Nissanka, Indrajith D. and Poojitha D. Yapa (2018). "Calculation of oil droplet size distribution in ocean oil spills: A review". In: *Marine Pollution Bulletin* 135.July, pp. 723–734.

NIST/SEMATECH (2012). *NIST/SEMATECH e-Handbook of Statistical Methods*.

NOAA (2012). *General NOAA Operational Modeling Environment (GNOME) Technical Documentation*. Tech. rep.

Nonomura, Taku, Hisaichi Shibata, and Ryoji Takaki (2019). *Extended-Kalman-filter-based dynamic mode decomposition for simultaneous system identification and denoising*. Vol. 14. 2. arXiv: 1805.01985.

Nordam, Tor et al. (2019a). "Numerical analysis of boundary conditions in a Lagrangian particle model for vertical mixing, transport and surfacing of buoyant particles in the water column". In: *Ocean Modelling* 136.January, pp. 107–119.

Nordam, Tor et al. (2019b). "On the use of random walk schemes in oil spill modelling". In: *Marine Pollution Bulletin* 146.April, pp. 631–638.

Ogata, Katsuhiko et al. (1995). *Discrete-time control systems*. Vol. 2. Prentice Hall Englewood Cliffs, NJ.

Pashna, Mohsen et al. (2020). "Autonomous multi-robot tracking system for oil spills on sea surface based on hybrid fuzzy distribution and potential field approach". In: *Ocean Engineering* 207.February 2019, p. 107238.

Peng, Hui Peng Hui et al. (2009). "Cooperative area search for multiple UAVs based on RRT and decentralized receding horizon optimization". In: *2009 7th Asian Control Conference*, pp. 298–303.

Pond, Stephen and George L Pickard (1983). *Introductory dynamical oceanography*. Gulf Professional Publishing.

Proctor, Roger, Roger A. Flather, and Alan J. Elliott (1994). "Modelling tides and surface drift in the Arabian Gulf-application to the Gulf oil spill". In: *Continental Shelf Research* 14.5, pp. 531–545.

Qu, Yaohong, Yintao Zhang, and Youmin Zhang (2015). "A UAV solution of regional surveillance based on pheromones and artificial potential field theory". In: *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 380–385.

Raanes, Patrick N., Marc Bocquet, and Alberto Carrassi (2019). "Adaptive covariance inflation in the ensemble Kalman filter by Gaussian scale mixtures". In: *Quarterly Journal of the Royal Meteorological Society* 145.718, pp. 53–75. arXiv: 1801.08474.

Ragi, Shankarachary and Edwin K P Chong (2013). "UAV path planning in a dynamic environment via partially observable markov decision process". In: *IEEE Transactions on Aerospace and Electronic Systems* 49.4, pp. 2397–2412.

Raimondi, Valentina et al. (2017). "Int J Appl Earth Obs Geoinformation Experimental tests and radiometric calculations for the feasibility of fluorescence LIDAR-based discrimination of oil spills from UAV". In: 61.April, pp. 46–54.

Rasmussen, Dorte (1985). "Oil Spill Modeling—a Tool for Cleanup Operations". In: *International Oil Spill Conference Proceedings* 1985.1, pp. 243–249.

Reed, Mark et al. (2000). "OSCAR2000: a multi-component 3-dimensional Oil Spill Contingency and Response model". In: *Arctic and Marine Oilspill Program Technical Seminar*.

Risken, Hannes (1996). "Fokker-planck equation". In: *The Fokker-Planck Equation*. Springer, pp. 63–95.

Roberge, Vincent, Mohammed Tarbouchi, and Gilles Labonte (2013). "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning". In: *IEEE Transactions on Industrial Informatics* 9.1, pp. 132–141.

Rozier, D et al. (2007). "A Reduced-Order Kalman Filter for Data Assimilation in Physical Oceanography". In: *SIAM Review* 49.3, pp. 449–465.

Schmid, Peter J. (2010). "Dynamic mode decomposition of numerical and experimental data". In: *Journal of Fluid Mechanics, Cambridge University Press (CUP),* 656, pp. 5–28.

Schouwenaars, Tom and Eric Feron (2004). "Decentralized Cooperative Trajectory Planning of Multiple Aircraft with Hard Safety Guarantees". In: *Control* August, AIAA 2004–5141.

Shahriari, Bobak et al. (2016). "Taking the human out of the loop: A review of Bayesian optimization". In: *Proceedings of the IEEE* 104.1, pp. 148–175.

Shang, Ke et al. (2015). "A GA-ACO hybrid algorithm for the multi-UAV mission planning problem". In: *14th International Symposium on Communications and Information Technologies, ISCIT 2014*, pp. 243–248.

Shannon, C. E. (1948). "Comment". In: *The Bell System Technical Journal* XXVII.3, pp. 379–423.

Smith, Stuart D. (1988). "Coefficients for sea surface wind stress, heat flux, and wind profiles as a function of wind speed and temperature". In: *Journal of Geophysical Research: Oceans* 93.C12, pp. 15467–15472.

Sonmez, Abdurrahim, Emre Kocyigit, and Emin Kugu (2015). "Optimal path planning for UAVs using Genetic Algorithm". In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 50–55.

Spaulding, Malcolm L. (2017). "State of the art review and future directions in oil spill modeling". In: *Marine Pollution Bulletin* 115.1-2, pp. 7–19.

Stam, Jos (2001). "A Simple Fluid Solver Based on the FFT". In: *Journal of Graphics Tools* 6, pp. 43–52.

— (2003). "Real-Time Fluid Dynamics for Games". In: *Proceedings of the Game Developer Conference* 18.11, p. 17.

Sun, Andrew K and Hugh H T Liu (2009). "Cooperative UAV Search for Moving Targets Using a Modified Diffusion Uncertainty Model". In: *AIAA Guidance, Navigation, and Control Conference* August.

Taylor, Publisher et al. (2003). "A multiphase oil spill model Un modèle multiphase de nappe d ' huile". In: *Journal Of Hydraulic Research* 41.2, pp. 115–125.

The Hong Kong Special Administrative Region Marine Department (2007). *Report of investigation into the Collision between the Hong Kong Registered ship "Hebei Spirit" and Korean Crane Barge "Samsung No. 1" on 7 December 2007*.

Thiébaut, Maxime and Alexei Sentchev (2016). "Tidal stream resource assessment in the Dover Strait (eastern English Channel)". In: *International Journal of Marine Energy* 16.February 2018, pp. 262–278.

Tkalich, Pavlo and Eng Soon Chan (2002). "Vertical mixing of oil droplets by breaking waves". In: *Marine Pollution Bulletin* 44.11, pp. 1219–1229.

Tong, Han et al. (2012). "Path planning of UAV based on Voronoi diagram and DPSO". In: *Procedia Engineering* 29, pp. 4198–4203.

Topouzelis, Konstantinos and Suman Singha (2016). "Oil spill detection : past and future trends". In: *ESA Living Planet Symposium* SP-740.

Tu, Jonathan H. et al. (2014). "On dynamic mode decomposition: Theory and applications". In: *Journal of Computational Dynamics* 1.2, pp. 391–421. arXiv: 1312.0041.

Turker, Tolgahan, Ozgur Koray Sahingoz, and Guray Yilmaz (2015). "2D path planning for UAVs in radar threatening environment using simulated annealing algorithm". In: *2015 International Conference on Unmanned Aircraft Systems, ICUAS 2015*, pp. 56–61.

UNCTAD (2016). *Review of Maritime Transport 2016*. Tech. rep., p. 34. arXiv: 1011.1669v3.

Vali, Mehdi et al. (2019). "Adjoint-based model predictive control for optimal energy extraction in waked wind farms". In: *Control Engineering Practice* 84.September 2018, pp. 48–62.

Vergassola, Massimo, Emmanuel Villermaux, and Boris I. Shraiman (2007). "'Infotaxis' as a strategy for searching without gradients". In: *Nature* 445.7126, pp. 406–409.

Visser, André W. (1997). "Using random walk models to simulate the vertical distribution of particles in a turbulent water column". In: *Marine Ecology Progress Series* 158.1, pp. 275–281.

Wang, Guoshi, Qiang Li, and Lejiang Guo (2010). "Multiple UAVs Routes Planning Based on Particle Swarm Optimization Algorithm". In: *2010 2nd International Symposium on Information Engineering and Electronic Commerce*, pp. 1–5.

Wang, Jinhua and Yongming Shen (2010). "Modeling oil spills transportation in seas based on unstructured grid, finite-volume, wave-ocean model". In: *Ocean Modelling* 35.4, pp. 332–344.

Wang, S. D., Y. M. Shen, and Y. H. Zheng (2005). "Two-dimensional numerical simulation for transport and fate of oil spills in seas". In: *Ocean Engineering* 32.13, pp. 1556–1571.

Wang, Zheng et al. (2015). "Multiple task planning based on TS algorithm for multiple heterogeneous unmanned aerial vehicles". In: *2014 IEEE Chinese Guidance, Navigation and Control Conference, CGNCC 2014*, pp. 630–635.

Wang, Zhi, Han Xiong Li, and Chunlin Chen (2020). "Reinforcement learning-based optimal sensor placement for spatiotemporal modeling". In: *IEEE Transactions on Cybernetics* 50.6, pp. 2861–2871.

Weatherly, Georges L. (1975). *A Numerical Study of Time-Dependent Turbulent Ekman Layers over Horizontal and Sloping Bottoms*.

Webb, A. and B. Fox-Kemper (2011). "Wave spectral moments and Stokes drift estimation". In: *Ocean Modelling* 40.3-4, pp. 273–288.

Weisstein, Eric W. (2017). *Random Walk–2-Dimensional*.

West, G. S. and C. J. Apelt (1982). "The effects of tunnel blockage and aspect ratio on the mean flow past a circular cylinder with Reynolds numbers between 104 and 105". In: *Journal of Fluid Mechanics* 114.January, pp. 361–377.

Wu, Jin (1975). "Wind-induced drift currents". In: *Journal of Fluid Mechanics* 68.01, pp. 49–70.

— (1980). "Wind-stress coefficients over sea surface near neutral conditions—A revisit". In: *Journal of Physical Oceanography* 10.5, pp. 727–740.

Xiao, Xiao et al. (2012). "A cooperative approach to multiple UAVs searching for moving targets based on a hybrid of virtual force and receding horizon". In: *IEEE International Conference on Industrial Informatics (INDIN)*, pp. 1228–1233.

Yan, Shuxue et al. (2018). "An AUV Adaptive Sampling Path Planning Method Based On Online Model Prediction". In: *IEEE 4th International Conference on Computer and Communications (ICCC)* Chengdu, C, pp. 2205–2211.

— (2019). "An AUV Adaptive Sampling Path Planning Method Based on Online Model Prediction". In: *IFAC-PapersOnLine* 52.21, pp. 323–328.

Ye, Xudong et al. (2019). "A simulation-based multi-agent particle swarm optimization approach for supporting dynamic decision making in marine oil spill responses". In: *Ocean and Coastal Management* 172.February, pp. 128–136.

Yi, Ting Hua and Hong Nan Li (2012). "Methodology developments in sensor placement for health monitoring of civil infrastructures". In: *International Journal of Distributed Sensor Networks* 2012.

Yildirim, B., C. Chryssostomidis, and G. E. Karniadakis (2009). "Efficient sensor placement for ocean measurements using low-dimensional concepts". In: *Ocean Modelling* 27.3-4, pp. 160–173.

Yu, Fangjie et al. (2017). "Calibration of backward-in-time model using drifting buoys in the East China Sea". In: *Oceanologia* 59.3, pp. 238–247.

Zammit Mangion, A., S. R. Anderson, and V. Kadirkamanathan (2011). "Exploration and control of stochastic spatiotemporal systems with mobile agents". In: *IFAC Proceedings Volumes (IFAC-PapersOnline)* 18.PART 1, pp. 4489–4494.

Zeigler, B P (1990). "High autonomy systems: concepts and models". In: *Proceedings [1990]. AI, Simulation and Planning in High Autonomy Systems.*

Zhang, Cheng and Hailong Pei (2015). "Oil spills boundary tracking using Universal Kriging and Model Predictive Control by UAV". In: *Proceedings of the World Congress on Intelligent Control and Automation (WCICA)* 2015-March.March, pp. 633–638.

Zhang, Yanmin et al. (2015). "The damping model for sea waves covered by oil films of a finite thickness". In: *Acta Oceanologica Sinica* 34.9, pp. 71–77.

Appendix A

Derivatives required for the Adjoint method

This section describes the analytical derivatives required for implementation of the Adjoint method and it may be useful to have Chapter 5 and Chapter 4 Section 4.6 for reference. This section begin with state and position derivatives of the system matrix. The constraint derivatives with respect to states and positions are then noted, followed by additional definitions for terms used within the prior derivatives. In this section, the notation $\overset{u}{\cdot}$ is dropped for the bounded and unbounded states and state trajectory of the uncertainty system, and they are referred to as $\vec{x}$ and $\vec{\mathscr{X}}$ for brevity, as there is no possibility of confusion. The notation $\boldsymbol{I}^{\vec{x}_k}$ represents an identity matrix for the dimensions of $\vec{x}_k$. To avoid tensor derivatives, evaluations of $\frac{\partial(\cdot)}{\partial \vec{p}}$ is performed per sensor and per movement direction of the selected sensor, for sensor $i \in [\![1, n_p]\!]$, at time t_k and a direction $d = x|y$, the derivative is with respect to $\vec{p}_{i_d}^{\,k} : \mathbb{N} \to \mathbb{R}$. Additionally, let the position vector of a sensor have the notation $\vec{p}_{i_v}^{\,k} : \mathbb{N} \to \mathbb{R}^2$. The Kronecker product is denoted by $\otimes$. The diag$(\vec{x})$ function here constructs a matrix with $\vec{x}$ along the diagonals, while Diag$(\boldsymbol{M})$ constructs a vector, containing the diagonal elements of $\boldsymbol{M}$.

This section is heavy in analytical tensor derivatives and the reader is advised to make use of the modern auto-differential products to describe their adjoint system (see Hu et al. 2019 for example). These auto-differential products were entering development at the beginning of this research.

Note that the derivative $\frac{\partial \boldsymbol{A}(\vec{x})\vec{x}}{\partial \vec{x}}$ is described by $\frac{\partial \boldsymbol{A}(\vec{x})\vec{x}}{\partial \vec{x}} = \frac{\partial \boldsymbol{A}(\vec{x})}{\partial \vec{x}}(\boldsymbol{I}^{\vec{x}} \otimes \vec{x}) + \boldsymbol{A}(\vec{x})\boldsymbol{I}^{\vec{x}}$, where $\boldsymbol{I}^{\vec{x}}$ is an identity matrix of a suitable dimension (the same number of rows) as $\vec{x}$.

A.0.1 System derivatives with respect to states.

Employ the notation $\mathscr{F}(\vec{\underline{x}}_{k+1}, \vec{\underline{x}}_k, \vec{\underline{p}}_k, \vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})_{\vec{\underline{x}}_m} = \mathscr{F}^{\vec{\underline{x}}_m}_{k+1,m} = \frac{\partial \mathscr{F}(...)}{\partial \vec{\underline{x}}_m}$, then for all $m \notin \{k, k+1\}$, there is a zero-valued derivative $\mathscr{F}^{\vec{\underline{x}}}_{k+1,m} = 0$. To determine the tensor derivative $\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial(\vec{\cdot})}$, without resorting to element-by-element calculus, the derivative is determined with respect to each scalar component of the vector $(\vec{\cdot})$. The full tensor derivative can then be constructed through manipulation of the matrix-by-scalar derivatives. Each matrix-by-scalar derivative, resulting in a matrix, is stacked underneath the last, from first state element to the last. This large and very sparse matrix, $\mathscr{D} \in \mathbb{R}^{(8n_x n_y)^2 \times 8 n_x n_y}$ is then reshaped, such that it is a combination of each stacked matrix with alternating rows. For example,

$$\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \\ B_{1,1} & B_{1,2} \\ B_{2,1} & B_{2,2} \\ C_{1,1} & C_{1,2} \\ C_{2,1} & C_{2,2} \end{bmatrix} \rightarrow \begin{bmatrix} A_{1,1} & B_{1,1} & C_{1,1} & A_{1,2} & B_{1,2} & C_{1,2} \\ A_{2,1} & B_{2,1} & C_{2,1} & A_{2,2} & B_{2,2} & C_{2,2} \end{bmatrix}. \tag{A.1}$$

Note that if $A = \frac{\partial \boldsymbol{F}}{\partial \vec{x}_1}$, $B = \frac{\partial \boldsymbol{F}}{\partial \vec{x}_2}$, $C = \frac{\partial \boldsymbol{F}}{\partial \vec{x}_3}$, where $\vec{x}_{1,2,3}$ are the 1st, 2nd and 3rd elements in $\vec{x}$, then the right-hand-side matrix in (A.1) is $\frac{\partial \boldsymbol{F}}{\partial \vec{x}_{1,2,3}}$ under the numerator partial derivative convention, using the definitions in Khang 2012. The derivatives of $\boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})$ are zero valued with respect to $\vec{\sigma}_x^2, \vec{\sigma}_y^2, \vec{s}_x, \vec{s}_y$. Recall the Differential-Algebraic Equation described by

$$\mathscr{F}(\vec{\underline{x}}_{k+1}, \vec{\underline{x}}_k, \vec{p}_k, \vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}}) = \underbrace{\left[-\underline{\boldsymbol{M}}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}}) \left(\boldsymbol{I} + \delta t \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) + \boldsymbol{I}^i_k\right] \vec{\underline{x}}_k}_{\mathscr{A}_k}$$
$$\underbrace{+ \left[\left(\boldsymbol{I} - \left(\boldsymbol{I}^i_{k+1} + \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right)\right)\right] \vec{\underline{x}}_{k+1}}_{\mathscr{C}_{k+1}} = 0. \tag{A.2}$$

The derivative with respect to the current-step states is $\mathscr{F}(...)_{\vec{x}_k}$, described by,

$$\mathscr{F}^{\vec{\underline{x}}}_{k+1,k} = -\frac{\partial \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{\underline{x}}_k} \left(\vec{\underline{x}}_{k+1} \otimes \boldsymbol{I}^{\vec{\underline{x}}_k}\right) + \left[-\underline{\boldsymbol{M}}_k \left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) + \boldsymbol{I}^i\right]$$
$$- \left(\frac{\partial \underline{\boldsymbol{M}}_k}{\partial \vec{\underline{x}}_k} \left(\left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) \otimes \boldsymbol{I}^{\vec{\underline{x}}_k}\right) + \underline{\boldsymbol{M}}_k \left(\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{\underline{x}}_k}\right)\right) \left(\vec{\underline{x}}_k \otimes \boldsymbol{I}^{\vec{\underline{x}}_k}\right). \tag{A.3}$$

The derivative with respect to the next states is $\mathscr{F}(...)_{\vec{x}_{k+1}}$, described by,

$$\mathscr{F}^{\vec{x}}_{k+1,k+1} = \left(\boldsymbol{I} - (\boldsymbol{I}^i + \underline{\boldsymbol{R}}_{k+1})\right) - \frac{\partial \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{\underline{x}}_{k+1}} \left(\vec{\underline{x}}_{k+1} \otimes \boldsymbol{I}^{\vec{\underline{x}}_{k+1}}\right) - \left(\frac{\partial \underline{\boldsymbol{M}}_k}{\partial \vec{\underline{x}}_{k+1}} \left(\left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) \otimes \boldsymbol{I}^{\vec{\underline{x}}_{k+1}}\right) + \underline{\boldsymbol{M}}_k \left(\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{\underline{x}}_{k+1}}\right)\right) \left(\vec{\underline{x}}_k \otimes \boldsymbol{I}^{\vec{\underline{x}}_{k+1}}\right). \quad \text{(A.4)}$$

Beginning with $\boldsymbol{Y}_k$, the derivative is non-zero valued for $\vec{u}_k$ and $\vec{v}_k$ only. Using MATLAB matrix indexing notation, the derivatives with respect to a selected by $i \in \{1, 2, ..., n_x n_y\}$ state of $\vec{u}_k$ and $\vec{v}_k$ are described by,

$$\frac{\partial \boldsymbol{Y}_k(i,:)}{\partial \vec{u}_k(i)} = -\boldsymbol{D}_x(i,:), \quad \frac{\partial \boldsymbol{Y}_k(i,:)}{\partial \vec{v}_k(i)} = -\boldsymbol{D}_y(i,:). \quad \text{(A.5)}$$

The derivatives of $\boldsymbol{T}_{k,u|v}$ with respect to $\vec{u}_k$ and $\vec{v}_k$ are also non-zero. The derivatives for $\boldsymbol{T}_{k,u}$ are described by,

$$\frac{\partial \boldsymbol{T}_{k,u}(i,:)}{\partial \vec{u}_k(i)} = \boldsymbol{I}(i,:) \circ \Big(c_d(\boldsymbol{D}_x(i,i) + \boldsymbol{D}_y(i,i))(2\vec{u}_k(i)^{\circ -1}) \circ ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} - c_d \vec{u}_k(i)^{\circ -2} \circ ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Big), \quad \text{(A.6)}$$

$$\frac{\partial \boldsymbol{T}_{k,u}(i,:)}{\partial \vec{u}_k(j \neq i)} = \boldsymbol{I}(i,:) \circ \Big(c_d(\boldsymbol{D}_x(i,j) - \boldsymbol{D}_y(i,j))(2\vec{u}_k(j)^{\circ -1}) \circ ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Big), \quad \text{(A.7)}$$

and

$$\frac{\partial \boldsymbol{T}_{k,u}(i,:)}{\partial \vec{v}_k(j)} = \boldsymbol{I}(i,:) \circ \Big(c_d(\boldsymbol{D}_x(i,j) - \boldsymbol{D}_y(i,j))(2\vec{u}_k(i)^{\circ -1}) \circ ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Big). \quad \text{(A.8)}$$

The derivatives for $\boldsymbol{T}_{k,v}$ are described by,

$$\frac{\partial \boldsymbol{T}_{k,v}(i,:)}{\partial \vec{v}_k(i)} =$$
$$\boldsymbol{I}(i,:)\circ\Bigg(c_d(\boldsymbol{D}_x(i,i)+\boldsymbol{D}_y(i,i))(2\vec{v}_k(i)^{\circ -1})\circ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5}$$
$$- c_d\vec{v}_k(i)^{\circ -2} \circ ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Bigg), \quad \text{(A.9)}$$

$$\frac{\partial \boldsymbol{T}_{k,v}(i,:)}{\partial \vec{v}_k(j \neq i)} =$$
$$\boldsymbol{I}(i,:)\circ\Bigg(c_d(\boldsymbol{D}_x(i,j)-\boldsymbol{D}_y(i,j))(2\vec{v}_k(j)^{\circ -1})\circ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Bigg), \quad \text{(A.10)}$$

and

$$\frac{\partial \boldsymbol{T}_{k,v}(i,:)}{\partial \vec{u}_k(j)} =$$
$$\boldsymbol{I}(i,:)\circ\Bigg(c_d(\boldsymbol{D}_x(i,j)-\boldsymbol{D}_y(i,j))(2\vec{v}_k(i)^{\circ -1})\circ((\boldsymbol{D}_x(i,:)\vec{u}_k - \boldsymbol{D}_y(i,:)\vec{v}_k) + (\boldsymbol{D}_y(i,:)\vec{u}_k + \boldsymbol{D}_x(i,:)\vec{v}_k))^{\circ 0.5} \Bigg). \quad \text{(A.11)}$$

The derivative of $\boldsymbol{Q}_{k+1_x}$ is non-zero for elements of $\vec{e}_{x_k}$ only, similar for $\boldsymbol{Q}_{k+1_y}$ and $\vec{e}_{y_k}$. The derivative of $\boldsymbol{Q}_{k+1_x}$ with respect to $\vec{e}_{x_k}(i)$ is described by,

$$\frac{\partial \boldsymbol{Q}_{k+1_x}(i,:)}{\partial \vec{e}_{x_k}(i)} = \boldsymbol{I}(i,:) \circ \Bigg(- \vec{e}_{x_k}(i)^{\circ -2} \circ \vec{E}_{k+1_x}(i) \Bigg), \quad \text{(A.12)}$$

with a similar expression in the vertical y direction.

To construct the total derivative with respect to a scalar variable in the state vector, the above derivative terms for the scalar variable are inserted into an empty matrix $\boldsymbol{Z} \in \mathbb{R}^{8n_xn_y \times 8n_xn_y}$, at their corresponding location. This is repeated for each element of the state vector, with each $\boldsymbol{Z}$ vertically concatenated to form $\boldsymbol{Q} \in \mathbb{R}^{(8n_xn_y)^2 \times 8n_xn_y}$, $\boldsymbol{Q}$ is then reshaped to form $\frac{\partial \boldsymbol{A}_k(\vec{\mathcal{Z}},\vec{\mathcal{P}})}{\partial \vec{x}_k} \in \mathbb{R}^{8n_xn_y \times (8n_xn_y)^2}$, a large and very sparse matrix. Note that a sparse representation of a matrix (only non-zero values are assigned) should be used at every stage to avoid memory constraints.

A.0.2 System derivatives with respect to sensor positions.

Employ the notation $\mathscr{F}(\vec{\underline{x}}_{k+1}, \vec{\underline{x}}_k, \vec{p}_k, \vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})_{\vec{p}_m} = \mathscr{F}^{\vec{p}}_{k+1,m} = \frac{\partial \mathscr{F}(...)}{\partial \vec{p}_m}$, then for all $m \notin \{k-1, k, k+1\}$, there is a zero-valued derivative $\mathscr{F}^{\vec{p}}_{k+1,m} = 0$.

The derivative with respect to the prior sensor positions is $\mathscr{F}(...)_{\vec{p}_{k-1}}$, described by,

$$\begin{aligned}\mathscr{F}^{\vec{p}}_{k+1,k-1} = &-\frac{\partial \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_{k-1}} \left(\vec{\underline{x}}_{k+1} \otimes \boldsymbol{I}^{\vec{p}}\right) \\ &- \left(\frac{\partial \underline{\boldsymbol{M}}_k}{\partial \vec{p}_{k-1}} \left(\left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) \otimes \boldsymbol{I}^{\vec{p}}\right) + \underline{\boldsymbol{M}}_k \left(\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_{k-1}}\right)\right) \left(\vec{\underline{x}}_k \otimes \boldsymbol{I}^{\vec{p}}\right). \quad \text{(A.13)}\end{aligned}$$

The derivative with respect to the current-step sensor positions is $\mathscr{F}(...)_{\vec{x}_k}$, described by,

$$\begin{aligned}\mathscr{F}^{\vec{p}}_{k+1,k} = &-\frac{\partial \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_k} \left(\vec{\underline{x}}_{k+1} \otimes \boldsymbol{I}^{\vec{p}}\right) + \\ &- \left(\frac{\partial \underline{\boldsymbol{M}}_k}{\partial \vec{p}_k} \left(\left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) \otimes \boldsymbol{I}^{\vec{p}}\right) + \underline{\boldsymbol{M}}_k \left(\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_k}\right)\right) \left(\vec{\underline{x}}_k \otimes \boldsymbol{I}^{\vec{p}}\right). \quad \text{(A.14)}\end{aligned}$$

The derivative with respect to the next sensor positions is $\mathscr{F}(...)_{\vec{x}_{k+1}}$, described by,

$$\begin{aligned}\mathscr{F}^{\vec{p}}_{k+1,k+1} = &-\frac{\partial \underline{\boldsymbol{R}}_{k+1}(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_{k+1}} \left(\vec{\underline{x}}_{k+1} \otimes \boldsymbol{I}^{\vec{p}}\right) \\ &- \left(\frac{\partial \underline{\boldsymbol{M}}_k}{\partial \vec{p}_{k+1}} \left(\left(\boldsymbol{I} + \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})\right) \otimes \boldsymbol{I}^{\vec{p}}\right) + \underline{\boldsymbol{M}}_k \left(\frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{p}_{k+1}}\right)\right) \left(\vec{\underline{x}}_k \otimes \boldsymbol{I}^{\vec{p}}\right). \quad \text{(A.15)}\end{aligned}$$

Define the matrix $\mathscr{D}_p \in \mathbb{R}^{n_\tau 8 n_x n_y \times n_p}$, that contains the derivative $\frac{\partial \boldsymbol{F}}{\partial \vec{p}_m}$, for $m \in \{k-1, k, k+1\}$,

in the following arrangement,

$$\frac{\partial \boldsymbol{F}}{\partial \vec{p}_m} = \begin{bmatrix} \frac{\partial \vec{x}_{k-1}}{\partial \vec{p}_{k-1}} & \frac{\partial \vec{x}_{k-1}}{\partial \vec{p}_k} & \frac{\partial \vec{x}_{k-1}}{\partial \vec{p}_{k+1}} \\ \frac{\partial \vec{x}_k}{\partial \vec{p}_{k-1}} & \frac{\partial \vec{x}_k}{\partial \vec{p}_k} & \frac{\partial \vec{x}_k}{\partial \vec{p}_{k+1}} \\ \frac{\partial \vec{x}_{k+1}}{\partial \vec{p}_{k-1}} & \frac{\partial \vec{x}_{k+1}}{\partial \vec{p}_k} & \frac{\partial \vec{x}_{k+1}}{\partial \vec{p}_{k+1}} \end{bmatrix} = \begin{bmatrix} \frac{\partial \vec{\sigma_x},_{k-1}}{\partial \vec{p}_1} & \frac{\partial \vec{\sigma_x},_{k-1}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_x},_{k-1}}{\partial \vec{p}_{k_p}} \\ \frac{\partial \vec{\sigma_y},_{k-1}}{\partial \vec{p}_1} & \frac{\partial \vec{\sigma_y},_{k-1}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_y},_{k-1}}{\partial \vec{p}_{k_p}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \vec{\sigma_x},_{k-1}}{\partial \vec{p}_1} & \frac{\partial \vec{s_y},_{k-1}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{s_y},_{k-1}}{\partial \vec{p}_{k_p}} \\ \frac{\partial \vec{\sigma_x},_{k}}{\partial \vec{p}_1} & \frac{\partial \vec{\sigma_x},_{k}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_x},_{k}}{\partial \vec{p}_{k_p}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \vec{s_y},_{k}}{\partial \vec{p}_1} & \frac{\partial \vec{s_y},_{k}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_x},_{k}}{\partial \vec{p}_{k_p}} \\ \frac{\partial \vec{\sigma_x},_{k+1}}{\partial \vec{p}_1} & \frac{\partial \vec{\sigma_x},_{k+1}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_x},_{k+1}}{\partial \vec{p}_{k_p}} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial \vec{s_y},_{k+1}}{\partial \vec{p}_1} & \frac{\partial \vec{s_y},_{k+1}}{\partial \vec{p}_2} & \cdots & \frac{\partial \vec{\sigma_x},_{k+1}}{\partial \vec{p}_{k_p}} \end{bmatrix} = \begin{bmatrix} \mathscr{F}^{\vec{p}}_{k-1,k-1} & \mathscr{F}^{\vec{p}}_{k-1,k} & \mathscr{F}^{\vec{p}}_{k-1,k+1} \\ \mathscr{F}^{\vec{p}}_{k,k-1} & \mathscr{F}^{\vec{p}}_{k,k} & \mathscr{F}^{\vec{p}}_{k,k+1} \\ \mathscr{F}^{\vec{p}}_{k+1,k-1} & \mathscr{F}^{\vec{p}}_{k+1,k} & \mathscr{F}^{\vec{p+1}}_{k+1,k+1} \end{bmatrix} \tag{A.16}$$

Begin with the derivative of $\boldsymbol{Y}_k(\vec{p}, \vec{p}_{k-1})$. The derivative with respect to the prior sensor positions is described by

$$\frac{\partial \boldsymbol{Y}_k(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}^{k-1}_{i_d}} = \operatorname{diag}\left(\frac{\left\| \delta\Omega - \vec{p}_{i_{k-1}} \right\|_d}{\left\| \delta\Omega - \vec{p}_{i_{k-1}} \right\|_2} \frac{k_s}{\delta t} \delta\left(\delta t - \frac{\left\| \delta\Omega - \vec{p}_{i_{k-1}} \right\|_2}{v_{\text{sensor}}} \right) \right), \tag{A.17}$$

where $\delta(\cdot)$ is the dirac delta function and the diag function operates on a vector, where each element corresponds to a location in $\partial\Omega$. The derivative with respect to the current-step sensor positions is described by

$$\frac{\partial \boldsymbol{Y}_k(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}^{k}_{i_d}} = \operatorname{diag}\left(\frac{\left\| \delta\Omega \cap (H_r \leq 1) - \vec{p}_{i_{k-1}} \right\|_d}{\left\| \delta\Omega \cap (H_r \leq 1) - \vec{p}_{i_{k-1}} \right\|_2} \frac{k_s}{\delta t} \delta\left(r - \left\| \delta\Omega \cap (H_r \leq 1) - \vec{p}_i \right\|_2 \right) \right). \tag{A.18}$$

The derivative of $\boldsymbol{Q}_{k+1_x}$ with respect to the prior sensor positions is described by,

$$\frac{\partial \boldsymbol{Q}_{k+1_x}}{\partial \vec{p}^{k-1}_{i_d}} = \operatorname{diag}\left((\vec{e}_x^{\circ -1}) \circ \frac{\partial \vec{E}_{k+1_x}}{\partial \vec{p}^{k-1}_{i_d}} \right), \tag{A.19}$$

where the further derivative is described by,

$$\frac{\partial \vec{E}_{k+1_x}}{\partial \vec{p}^{k-1}_{i_d}} = \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\left(\boldsymbol{I} - \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k) \boldsymbol{C}_{\text{KF}_{k+1}}(\vec{p}_{k+1}) \right) \left((\Lambda^{\frac{\delta t}{\Delta t}}) \frac{\partial \boldsymbol{P}_{\text{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}^{k-1}_{i_d}} (\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_{\text{KF}} \right) \right) \boldsymbol{\Psi}^*_u \right) \tag{A.20}$$

The term $\boldsymbol{R}_{\mathrm{KF}} \in \mathbb{R}^{2n_p \times 2n_p}$ is the measurement covariance matrix only for the current-time sensor measurements. The derivative $\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^{k-1}}$ is described in (A.43), and additional terms are described in Section A.0.4.

The derivative of $\boldsymbol{Q}_{k+1_x}$ with respect to the current-time sensor positions is described by,

$$\frac{\partial \boldsymbol{Q}_{k+1_x}}{\partial \vec{p}_{i_d}^{k}} = \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\left(\boldsymbol{I} - \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)\boldsymbol{C}_{\mathrm{KF}_{k+1}}(\vec{p}_{k+1})\right)\left((\Lambda^{\frac{\delta t}{\Delta t}})\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^{k}}(\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_{\mathrm{KF}}\right)\right)\boldsymbol{\Psi}\right.$$
$$\left. + \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\left(\boldsymbol{I} - \frac{\partial \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)}{\partial \vec{p}_{i_d}^{k}}\boldsymbol{C}_{\mathrm{KF}_{k+1}}(\vec{p}_{k+1})\right)\left((\Lambda^{\frac{\delta t}{\Delta t}})\boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}(\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_{\mathrm{KF}}\right)\right)\boldsymbol{\Psi}_u^*\right), \tag{A.21}$$

where the derivatives $\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^{k}}$ and $\frac{\partial \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)}{\partial \vec{p}_{i_d}^{k}}$ are described in (A.37) and (A.45) respectively.

The derivative of $\boldsymbol{Q}_{k+1_x}$ with respect to the next sensor positions is described by,

$$\frac{\partial \boldsymbol{Q}_{k+1_x}}{\partial \vec{p}_{i_d}^{k+1}} = \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\left(\boldsymbol{I} - \frac{\partial \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)}{\partial \vec{p}_{i_d}^{k+1}}\boldsymbol{C}_{\mathrm{KF}_{k+1}}(\vec{p}_{k+1})\right)\left((\Lambda^{\frac{\delta t}{\Delta t}})\boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}(\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_{\mathrm{KF}}\right)\right)\right.$$
$$\left. + \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\left(\boldsymbol{I} - \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)\frac{\partial \boldsymbol{C}_{\mathrm{KF}_{k+1}}(\vec{p}_{k+1})}{\partial \vec{p}_{i_d}^{k+1}}\right)\left((\Lambda^{\frac{\delta t}{\Delta t}})\boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}(\Lambda^{\frac{\delta t}{\Delta t}})^* + \boldsymbol{R}_{\mathrm{KF}}\right)\right)\boldsymbol{\Psi}_u^*\right), \tag{A.22}$$

where the derivative $\frac{\partial \boldsymbol{L}_{k+1}(\vec{p}_{k+1}, \vec{p}_k)}{\partial \vec{p}_{i_d}^{k+1}}$ is described in (A.39), while $\frac{\partial \boldsymbol{C}_{\mathrm{KF}_{k+1}}(\vec{p}_{k+1})}{\partial \vec{p}_{i_d}^{k+1}}$ is calculated by the finite difference of sensor matrices with peturbed locations of sensor i in direction d.

The above derivatives are used to form $\frac{\partial \boldsymbol{A}_k(\vec{\mathscr{Z}}, \vec{\mathscr{P}})}{\partial \vec{p}_{i_d}^{m}}$, which are then vertically concatenated and reshaped to form $\frac{\partial \boldsymbol{A}_k(\vec{\mathscr{Z}}, \vec{\mathscr{P}})}{\partial \vec{p}_m}$. There are similar derivative expressions for $\boldsymbol{Q}_{k+1_y}$.

The derivative $\frac{\partial J}{\partial \vec{x}_k}$ is described by,

$$\frac{\partial J}{\partial \vec{x}_k} = \begin{bmatrix} \vec{E}(\vec{x},t) \circ k_\chi^2 \sigma_y^2 \\ \vec{E}(\vec{x},t) \circ k_\chi^2 \sigma_x^2 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{A.23}$$

where $\vec{E}(\vec{x},t) \in \mathbb{R}^{n_x n_y}$ is a vector of the uncertainty minimisation weighting $E(\vec{x},t)$ across the spatial domain.

The further derivative $\frac{\partial J}{\partial \vec{p}_k}$ is described by,

$$\frac{\partial J}{\partial \vec{p}_k} = \left[\frac{\partial c(t,\vec{p}_k,\vec{p}_{k-1})}{\partial \vec{p}_{1_x}^k} + \frac{\partial c(t,\vec{p}_{k+1},\vec{p}_k)}{\partial \vec{p}_{1_x}^k} \quad \frac{\partial c(t,\vec{p}_k,\vec{p}_{k-1})}{\partial \vec{p}_{1_y}^k} + \frac{\partial c(t,\vec{p}_{k+1},\vec{p}_k)}{\partial \vec{p}_{1_y}^k} \quad \dots \quad \frac{\partial c(t,\vec{p}_k,\vec{p}_{k-1})}{\partial \vec{p}_{k_{s,x}}^k} + \frac{\partial c(t,\vec{p}_k,\vec{p}_k)}{\partial \vec{p}_{k_{s,x}}^{k+1}} \quad \frac{\partial c(t,\vec{p},\vec{p}_{k-1})}{\partial \vec{p}_{k_{s,y}}^{k+1}} + \frac{\partial c(t,\vec{p},\vec{p}_k)}{\partial \vec{p}_{k_{s,y}}^k} \right], \tag{A.24}$$

where the complete derivative of the cost function J with respect to the sensor position trajectory for a 5-time-step horizon is described by

$$\frac{\partial J}{\partial \vec{\mathscr{P}}} = \begin{bmatrix} \frac{\partial J}{\partial \vec{p}_{k-2}} & \frac{\partial J}{\partial \vec{p}_{k-1}} & \frac{\partial J}{\partial \vec{p}_k} & \frac{\partial J}{\partial \vec{p}_{k+1}} & \frac{\partial J}{\partial \vec{p}_{k+2}} \end{bmatrix} \tag{A.25}$$

A.0.3 Constraint derivatives.

The constraint term $c(t,\vec{p}_k,\vec{p}_{k-1})$ is the sum of three components. The first, $V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})$, is described by

$$V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}}) = \sum_{i=1}^{i=n_p} \left[\left\| \vec{p}_{i_v}^k - \vec{p}_{i_v}^{k-1} \right\|_2 H\left(\frac{\left\| \vec{p}_{i_v}^k - \vec{p}_{i_v}^{k-1} \right\|_2}{v_{\text{sensor}}} - \delta t \right) \right], \tag{A.26}$$

and represents the velocity constraints of the sensor by incurring additional penalty when the sensor position is in violation of these constraints, in addition to the loss of sensing due to the Heaviside functions in (4.86a) and (4.86b). The derivative $\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k}_{i_d}}$ is described by,

$$\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k}_{i_d}} = \frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2} H\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right) + \left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2 \left(\frac{1}{v_{\text{sensor}}}\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}\right)\delta\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right), \quad \text{(A.27)}$$

which simplifies to,

$$\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k}_{i_d}} = \frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2} H\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right) + \frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{v_{\text{sensor}}}\delta\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right). \quad \text{(A.28)}$$

The derivative $\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k-1}_{i_d}}$ is described by,

$$\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k-1}_{i_d}} = -\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2} H\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right) - \left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2 \left(\frac{1}{v_{\text{sensor}}}\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}\right)\delta\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right), \quad \text{(A.29)}$$

which simplifies to

$$\frac{\partial V(\vec{p}_k,\vec{p}_{k-1},v_{\text{sensor}})}{\partial \vec{p}^{\,k-1}_{i_d}} = -\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2} H\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right) - \frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_d}{v_{\text{sensor}}}\delta\left(\frac{\left\|\vec{p}^{\,k}_{i_v}-\vec{p}^{\,k-1}_{i_v}\right\|_2}{v_{\text{sensor}}}-\delta t\right). \quad \text{(A.30)}$$

The second component, $D_m(\vec{p}_k)$, is the Euclidean distance from each sensor to the closest location of $E(t) > 0$ in $\partial\Omega$, summated for all sensors. Define a set of $n_e \in \mathbb{N}$ points, $\mathscr{E}_m : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^{n_m \times 2}$, where $\mathscr{E}_m$ is described by $\mathscr{E}_m = (\partial\Omega \cap E(t) > 0)$. Then $\mathscr{E}_m$ is the set of grid cells where $E(t) > 0$. Let $l_m \in \mathbb{N} \cap [1, n_m]$ be defined such that it minimises the following: $\min_{l_m} \left\|\mathscr{E}(l_m)_m - \vec{p}^{\,k}_{i_v}\right\|_2$. The term $\mathscr{E}(l)_m$ is the closest location of $E(t) > 0$ to sensor i. The

constraint term component is then described by,

$$D_m(\vec{p}_k) = \sum_{i=1}^{i=n_p} \left[\left\| \mathscr{E}(l_m)_m - \vec{p}_{i_v}^k \right\|_2 \right], \tag{A.31}$$

and incurs additional penalty when the sensor position is not in a region of interest, providing incentive for sensors to move towards the region of interest in the absence of other information. When calculating derivatives, there is an assumption of a constant $\mathscr{E}(l_m)_m$. This is valid as a minute perturbation in sensor position is unlikely to alter the closest location of $E(t) > 0$. The derivative $\frac{\partial D_m(\vec{p}_k)}{\partial \vec{p}_{i_d}^k}$ is defined by

$$\frac{\partial D_m(\vec{p}_k)}{\partial \vec{p}_{i_d}^k} = \frac{-\left\| \mathscr{E}(l_m)_m - \vec{p}_{i_v}^k \right\|_d}{\left\| \mathscr{E}(l_m)_m - \vec{p}_{i_v}^k \right\|_2} \tag{A.32}$$

The third component, $D_e(\vec{p}_k)$, is the Euclidean distance from each sensor to the closest permissible location in $\partial\Omega$, summated for all sensors. Define a set of points $\mathscr{E}_e : \partial\Omega \times \mathbb{R}_+ \to \mathbb{R}^{n_e \times 2}$ as $n_e \in \mathbb{N}$ permissible locations in $\partial\Omega$. Let $l_e \in \mathbb{N} \cap [1, n_e]$ now be defined such that it minimises the following: $\min_{l_e} \left\| \mathscr{E}(l_e)_e - \vec{p}_{i_v}^k \right\|_2$. The term $\mathscr{E}(l_e)_e \in \mathbb{R}^2$ is the closest permissible location to sensor i. The constraint term component is then described by,

$$D_e(\vec{p}_k) = \sum_{i=1}^{i=n_p} \left[\left\| \mathscr{E}(l_e)_e - \vec{p}_{i_v}^k \right\|_2 \right], \tag{A.33}$$

and incurs additional penalty when the sensor position is within an excluded region, providing incentive for sensors to move towards a permissible region. The derivative $\frac{\partial D_e(\vec{p}_k)}{\partial \vec{p}_{i_d}^k}$ is defined by

$$\frac{\partial D_e(\vec{p}_k)}{\partial \vec{p}_{i_d}^k} = \frac{-\left\| \mathscr{E}(l_e)_e - \vec{p}_{i_v}^k \right\|_d}{\left\| \mathscr{E}(l_e)_e - \vec{p}_{i_v}^k \right\|_2}. \tag{A.34}$$

A.0.4 Estimation error derivatives.

To avoid tensor derivatives, evaluation of $\frac{\partial(\cdot)}{\partial \vec{p}}$ is performed per sensor and per movement direction of the selected sensor. The individual derivatives are then suitably utilised later. This approach is used repeatedly throughout. The derivatives of the variance sources for sensor

$i \in [1, n_p]$, at time t and a direction $d = x|y$, $\vec{p}_{i_d}^k$, are described by,

$$\frac{\partial \vec{E}_{\mathrm{KF}_x}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^k} = \boldsymbol{G}_x \operatorname{diag}\left(\boldsymbol{\Psi}_u \left(\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^k}\right) \boldsymbol{\Psi}_u^*\right), \tag{A.35a}$$

and

$$\frac{\partial \vec{E}_{\mathrm{KF}_y}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^k} = \boldsymbol{G}_y \operatorname{diag}\left(\boldsymbol{\Psi}_v \left(\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^k}\right) \boldsymbol{\Psi}_v^*\right), \tag{A.35b}$$

respectively. In the time varying Kalman filter, $\boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}$ is a function of the Kalman gain $\boldsymbol{L}(\vec{p}, \vec{p}_{k-1})$, the sensor matrix $\boldsymbol{C}_{\mathrm{KF}}(\vec{p})$ and the prior error covariance matrix, predicted from the previous time-step for this time-step, $\boldsymbol{R}_m(\vec{p}_{k-1}) = \boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1}, \vec{p}_{k-1})_{k|k-1}$. The posterior estimated error covariance $\boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}$ is defined by

$$\boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k} = (\boldsymbol{I} - \boldsymbol{L}(\vec{p}, \vec{p}_{k-1})\boldsymbol{C}_{\mathrm{KF}}(\vec{p}))\boldsymbol{R}_m(\vec{p}_{k-1})^*. \tag{A.36}$$

The derivative is defined by

$$\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}_k, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^k} = -\left(\boldsymbol{C}_{\mathrm{KF}}(\vec{p})\frac{\partial \boldsymbol{L}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^k} + \frac{\partial \boldsymbol{C}_{\mathrm{KF}}(\vec{p})}{\partial \vec{p}_{i_d}^k}\boldsymbol{L}(\vec{p}, \vec{p}_{k-1})\right)\boldsymbol{R}_m(\vec{p}_{k-1})^*, \tag{A.37}$$

with the Kalman gain decribed

$$\boldsymbol{L}(\vec{p}, \vec{p}_{k-1}) = (\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{C}_{\mathrm{KF}}(\vec{p})^*)(\boldsymbol{C}_{\mathrm{KF}}(\vec{p})\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{C}_{\mathrm{KF}}(\vec{p})^* + \boldsymbol{R}_{\mathrm{KF}})^{-1}. \tag{A.38}$$

Let $\boldsymbol{K}_{\mathrm{KF}} = (\boldsymbol{C}_{\mathrm{KF}}(\vec{p})\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{C}_{\mathrm{KF}}(\vec{p})^* + \boldsymbol{R}_{\mathrm{KF}})$, then,

$$\frac{\partial \boldsymbol{L}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^k} = (\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{D}_{\mathrm{KF}}^*)\boldsymbol{K}_{\mathrm{KF}}^{-1} - \boldsymbol{K}_{\mathrm{KF}}^{-1}(\boldsymbol{C}_{\mathrm{KF}}(\vec{p})\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{D}_{\mathrm{KF}}^* + \boldsymbol{D}_{\mathrm{KF}}\boldsymbol{R}_m(\vec{p}_{k-1})\boldsymbol{C}_{\mathrm{KF}}(\vec{p}))\boldsymbol{K}_{\mathrm{KF}}^{-1}, \tag{A.39}$$

The sensor matrix derivative $\boldsymbol{D}_{\mathrm{KF}} = \frac{\partial \boldsymbol{C}_{\mathrm{KF}}(\vec{p})}{\partial \vec{p}_{i_d}^k}$ is calculated by the finite difference of sensor matrices with perturbed locations of sensor i in direction d. The prior error covariance matrix $\boldsymbol{R}_m(\vec{p}_{k-1})$ is described by,

$$\boldsymbol{R}_m(\vec{p}_{k-1}) = \left(\boldsymbol{\Lambda}\boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1}, \vec{p}_{k-2})_{k-1|k-1}\boldsymbol{\Lambda}^* + \boldsymbol{R}_e\right). \tag{A.40}$$

Recall that $\boldsymbol{R}_e$ is the covariance matrix of the reduced order model states. As before, although the description of $\boldsymbol{R}_m(\vec{p}_{k-1})$ has a dependency upon $\vec{p}_{k-2}$, this is omitted in other notation in the interest of brevity. It is clear though, that as a sequential system, each iteration of the time-varying Kalman Filter is dependent upon the one before. The description of $\boldsymbol{R}_m(\vec{p}_{k-1})$ has the derivatives,

$$\frac{\partial \boldsymbol{R}_m(\vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,t}} = 0 \tag{A.41}$$

and

$$\frac{\partial \boldsymbol{R}_m(\vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}} = \boldsymbol{\Lambda} \frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}_{k-1}, \vec{p}_{k-2})_{k-1|k-1}}{\partial \vec{p}_{i_d}^{\,k-1}} \boldsymbol{\Lambda}^*. \tag{A.42}$$

In (A.42), the derivative term is equal to (A.37), but at the prior estimation step, $k-1$. The derivative of $\boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}$ with respect to the prior sensor positions is described by,

$$\frac{\partial \boldsymbol{P}_{\mathrm{r}}(\vec{p}, \vec{p}_{k-1})_{k|k}}{\partial \vec{p}_{i_d}^{\,k-1}} = \frac{\partial (I - \boldsymbol{L}(\vec{p}, \vec{p}_{k-1}) \boldsymbol{C}_{\mathrm{KF}}(\vec{p}))}{\partial \vec{p}_{i_d}^{\,k-1}} \boldsymbol{R}_m(\vec{p}_{k-1}) + (I - \boldsymbol{L}(\vec{p}, \vec{p}_{k-1}) \boldsymbol{C}_{\mathrm{KF}}(\vec{p})) \frac{\partial \boldsymbol{R}_m(\vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}}, \tag{A.43}$$

where,

$$\frac{\partial (I - \boldsymbol{L}(\vec{p}, \vec{p}_{k-1}) \boldsymbol{C}_{\mathrm{KF}^*}(\vec{p}))}{\partial \vec{p}_{i_d}^{\,k-1}} = -\frac{\partial \boldsymbol{L}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}} \boldsymbol{C}_{\mathrm{KF}^*}(\vec{p}), \tag{A.44}$$

and

$$\frac{\partial \boldsymbol{L}(\vec{p}, \vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}} = \left(\frac{\partial \boldsymbol{R}_m(\vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}} \boldsymbol{C}_{\mathrm{KF}}(\vec{p})^* \right) \boldsymbol{K}_{\mathrm{KF}}^{-1} - \boldsymbol{K}_{\mathrm{KF}}^{-1} \left(\boldsymbol{C}_{\mathrm{KF}}(\vec{p}) \frac{\partial \boldsymbol{R}_m(\vec{p}_{k-1})}{\partial \vec{p}_{i_d}^{\,k-1}} \boldsymbol{C}_{\mathrm{KF}}(\vec{p})^* \right) \boldsymbol{K}_{\mathrm{KF}}^{-1}. \tag{A.45}$$

A.0.5 State transition derivative

Recall define the state transition function,

$$\mathscr{G}(\underline{\vec{x}}_k, \vec{p}_k, \underline{\vec{\mathscr{X}}}, \vec{\mathscr{P}}) \coloneqq \boldsymbol{I} \underline{\vec{x}}_k + \boldsymbol{A}_k(\underline{\vec{\mathscr{X}}}, \vec{\mathscr{P}}) \underline{\vec{x}}_k = \vec{x}_{k+1}, \tag{A.46}$$

with the derivative, with respect to a vector $\vec{k}_m \in \mathbb{R}^{8n_x \times n_y}$, described by

$$\frac{\partial \mathscr{G}(\underline{\vec{x}}_k, \vec{p}_k, \underline{\vec{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{k}_m} = \boldsymbol{I} \frac{\partial \underline{\vec{x}}_k}{\partial \vec{k}_m} + \frac{\partial \boldsymbol{A}_k(\underline{\vec{\mathscr{X}}}, \vec{\mathscr{P}})}{\partial \vec{k}_m} \left(\boldsymbol{I}^{\vec{k}_m} \otimes \underline{\vec{x}}_k \right) + \boldsymbol{A}_k(\underline{\vec{\mathscr{X}}}, \vec{\mathscr{P}}) \frac{\partial \underline{\vec{x}}_k}{\partial \vec{k}_m}, \tag{A.47}$$

and the derivative with respect to the position vector $\vec{p}_k$ described by,

$$\frac{\partial \mathscr{G}(\vec{\underline{x}}_k, \vec{p}_k, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{p}_k} = \frac{\partial \boldsymbol{A}_k(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{p}_k} \left(\boldsymbol{I}^{\vec{p}_k} \otimes \vec{\underline{x}}_k \right). \tag{A.48}$$

Limit derivatives:

Now it is necessary to define the derivatives of the limit matrices, $\underline{\boldsymbol{M}}_n$ and $\underline{\boldsymbol{R}}_{n+1}$. The general derivatives descriptions are given, for $\vec{k}_m$, with the difference upon the value of $m \in [0, N_t]$ being that $\frac{\partial \vec{\underline{x}}_{n+1}}{\partial \vec{k}_m} = \boldsymbol{I}$ for $m = n+1$ and $\vec{k} = \vec{\underline{x}}$, or $\frac{\partial \vec{\underline{x}}_{n+1}}{\partial \vec{k}_m} = 0$ otherwise. Recall that

$$\underline{\boldsymbol{M}}_n(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}}) = \operatorname{diag}\left(\vec{\underline{x}}_{n+1} \circ \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})^{\circ -1} \right). \tag{A.49}$$

Recall that $\vec{k}_m \in \mathbb{R}^{n_m}$ and $\boldsymbol{E}_n(\vec{\mathscr{X}}, \vec{\mathscr{P}}) \in \mathbb{R}^{N_s \times N_s}$, $\boldsymbol{A}_n(\vec{\mathscr{X}}, \vec{\mathscr{P}}) \in \mathbb{R}^{N_s \times N_s}$ and $\boldsymbol{B}_n(\vec{\mathscr{X}}, \vec{\mathscr{P}}) \in \mathbb{R}^{N_s \times N_p}$. The derivative $\frac{\partial \underline{\boldsymbol{M}}_n(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{k}_m}$ is defined by,

$$\frac{\partial \underline{\boldsymbol{M}}_n(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{k}_m} = \left(\boldsymbol{I}_{N_s} \otimes 1^*_{N_m} \right) \circ \left(\frac{\partial \left(\vec{\underline{x}}_{n+1} \circ \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})^{\circ -1} \right)}{\partial \vec{k}_m} \otimes 1^*_{N_s} \right), \tag{A.50}$$

where

$$\begin{aligned} \frac{\partial \left(\vec{\underline{x}}_{n+1} \circ \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})^{\circ -1} \right)}{\partial \vec{k}_m} = \Bigg[& \left(\frac{\partial \vec{\underline{x}}_{n+1}}{\partial \vec{k}_m} \operatorname{diag}\left(\vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})^{\circ -1} \right) \right) \\ & + \left(\operatorname{diag}\left(\vec{\underline{x}}_{n+1} \right) \left(- \frac{\partial \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{k}_m} \operatorname{diag}\left(\vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})^{\circ -2} \right) \right) \right) \Bigg]. \end{aligned} \tag{A.51}$$

For the right-hand-side limit, to be applied when a state is zero-valued, recall that

$$\underline{\boldsymbol{R}}_{n+1}(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}}) = \operatorname{diag}\left(1 - \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}}) \circ (\vec{\underline{x}}_{n+1})^{\circ -1} \right). \tag{A.52}$$

The derivative $\frac{\partial \underline{\boldsymbol{R}}_{n+1}(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{k}_m}$ is defined by,

$$\frac{\partial \underline{\boldsymbol{R}}_{n+1}(\vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}})}{\partial \vec{k}_m} = \left(\boldsymbol{I}_{N_s} \otimes 1^*_{N_m} \right) \circ \left(\frac{\partial \left(1 - \vec{\mathscr{G}}(\vec{\underline{x}}_n, \vec{p}_n, \vec{\underline{\mathscr{X}}}, \vec{\underline{\mathscr{P}}}) \circ (\vec{\underline{x}}_{n+1})^{\circ -1} \right)}{\partial \vec{k}_m} \otimes 1^*_{N_s} \right), \tag{A.53}$$

where

$$\frac{\partial\left(1-\vec{\mathscr{G}}(\underline{\vec{x}}_n,\vec{p}_n,\underline{\vec{\mathscr{X}}},\vec{\mathscr{P}})\circ(\underline{\vec{x}}_{n+1})^{\circ-1}\right)}{\partial\vec{k}_m}=\Bigg[-\left(\frac{\partial\vec{\mathscr{G}}(\underline{\vec{x}}_n,\vec{p}_n,\underline{\vec{\mathscr{X}}},\vec{\mathscr{P}})}{\partial\vec{k}_m}\operatorname{diag}\left((\underline{\vec{x}}_{n+1})^{\circ-1}\right)\right)$$
$$+\left(-\operatorname{diag}\left(\vec{\mathscr{G}}(\underline{\vec{x}}_n,\vec{p}_n,\underline{\vec{\mathscr{X}}},\vec{\mathscr{P}})\right)\left(\frac{\partial\underline{\vec{x}}_{n+1}}{\partial\vec{k}_m}\operatorname{diag}\left((\underline{\vec{x}}_{n+1})^{\circ-2}\right)\right)\right)\Bigg]. \quad \text{(A.54)}$$