On Coupling Resolved and Unresolved Physical Processes in Finite Element Discretisations of Geophysical Fluids

Thomas Matthew Bendall

Department of Mathematics

Imperial College London

March 2020

A Thesis Presented for the Degree of Doctor of Philosophy in Mathematics

and the Diploma of Imperial College

DECLARATION

I declare that this dissertation and the research presented within it are the products of my own work. Work done jointly with others is made clear and the ideas and quotations from the work of others, published or otherwise, are fully acknowledged.

Signed: Thomas Bendall

COPYRIGHT

Abstract

On Coupling Resolved and Unresolved Physical Processes in Finite Element Discretisations of Geophysical Fluids

At the heart of modern numerical weather forecasting and climate modelling lie simulations of two geophysical fluids: the atmosphere and the ocean. These endeavours rely on numerically solving the equations that describe these fluids. A key challenge is that the fluids contain motions spanning a range of scales. As the small-scale processes (unresolved by the numerical model) affect the resolved motions, they need to be described in the model, which is known as parametrisation. One major class of methods for numerically solving such partial differential equations is the finite element method. This thesis focuses on the coupling of such parametrised processes to the resolved flow within finite element discretisations. Four sets of research are presented, falling under two main categories.

The first is the development of a compatible finite element discretisation for use in numerical weather prediction models, so as to avoid the bottleneck in computational scalability associated with the convergence at the poles of latitude-longitude grids. We present a transport scheme for use with the lowest-order function spaces in such a compatible finite element method, which is motivated by the coupling of the resolved and unresolved processes within the model. This then facilitates the use of the lower-order spaces within Gusto, a toolkit for studying such compatible finite element discretisations. Then, we present a compatible finite element discretisation of the moist compressible Euler equations, parametrising the unresolved moist processes. This is a major step in the development of Gusto, extending it to describe its first unresolved processes.

The second category with which this thesis is concerned is the stochastic variational framework presented by Holm [Variational principles for stochastic fluid dynamics, *P. Roy. Soc. A-Math. Phy.* 471 (*2176*), (2015)]. In this framework, the effect of the unresolved processes and their uncertainty is expressed through a stochastic component to the advecting velocity. This framework ensures the circulation theorem is preserved by the stochastic equations. We consider the application of this formulation to two simple geophysical fluid models. First, we discuss the statistical properties of an enstrophy-preserving finite element discretisation of the stochastic quasi-geostrophic equation. We find that the choice of discretisation and the properties that it preserves affects the statistics of the solution. The final research presented is a finite element discretisation of the stochastic Camassa-Holm equation, which is used to numerically investigate the formation of ‘peakons’ within this set-up, finding that they do still always form despite the noise’s presence.

Acknowledgments

Throughout the last four years my supervisors, Professor Colin Cotter, Dr Jemma Shipton, Dr Ben Shipway and Professor Darryl Holm, have offered me so much time, suggestions, and encouragement. Their support has been invaluable and I am very grateful for all the generosity and kindness they have shown towards me.

I began my studies at Imperial College without any experience of programming in Python or in a group environment, but the feedback and help of Jemma and the other developers of Gusto has been crucial in helping me to learn to code (and hopefully to code well). Ben's guidance in directing the moisture part of the project was far more vital than he'll realise, while Darryl's creativity and infectious enthusiasm can make even dreary maths more fun. In particular I'd like to thank Colin for meeting with me for an hour every week, providing so many of the ideas used in this research and having patience with my endless stream of naïve questions.

My opportunity to perform this research has only been made possible through the funding of the Engineering and Physical Sciences Research Council (EPSRC) and the Mathematics of Planet Earth Centre for Doctoral Training (MPE CDT) at Imperial College London and the University of Reading. The strength of the CDT is that students study in cohorts, providing us with an instant group of friends with whom we could show solidarity through the trials and tribulations of the PhD student experience. We all have gratitude for the CDT staff, and especially Anna Radomska and Craig Smith, whose tireless work was vital in ensuring the smooth running of the CDT.

I would also like to thank David Ham, Lawrence Mitchell and especially Thomas Gibson for answering so many of my Firedrake-related queries, no matter how banal or badly-worded. The many useful comments and suggestions of my examiners, Professor Beth Wingate and Professor Matthew Piggott were important in revising the thesis.

Without the encouragement of my parents and my undergraduate tutor Dr Chris Lester, I would have never thought myself capable of completing post-graduate studies, and I also owe my friends, who tolerated the moments when I was grumpy and my explanations of some of the more tedious details of my research.

Finally to Anna, who has probably suffered as much as me through this process – thank you for your unwavering support and patience, and of course your hours of proofreading. This work is dedicated to you.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

Introduction and Background

1 Introduction

The key ingredient underpinning modern weather forecasting and climate modelling is a mathematical understanding of the atmosphere and the ocean. These are the principal focus of the branch of mathematics known as geophysical fluid dynamics, which studies the evolution of natural fluids occurring on Earth and other planets. Other examples of geophysical fluids include lava flows, glaciers and the Earth's mantle, although sometimes these are considered to be the subject of other areas of science, rather than of geophysical fluid dynamics. The key features of geophysical fluids are the planet's gravitational field, which can cause the fluid to become stratified with denser fluid at the bottom; and the rotation of the planet, which gives rise to centrifugal and Coriolis forces when describing the fluid in a reference frame rotating with the planet. An overview and motivation for the study of geophysical fluids can be found in [1].

Although there are many ways to mathematically describe the evolution of a fluid, common variable choices are the velocity $\boldsymbol{v}$ of the fluid and the mass density ρ, which are fields dependent on spatial location $\boldsymbol{x}$ and time t. Assumptions about mass conservation and the application of Newton's Second Law lead to partial differential equations for $\boldsymbol{v}$ and ρ. The system of equations may be closed by additional equations, often a thermodynamic equation providing conservation of energy and an equation of state. A property common to these fluids models is the presence of non-linearity in the momentum equation, for instance many formulations contain a $(\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v}$ term. The significance of this non-linear term is that fluid motions on different length and time scales interact with one another. For the atmosphere and ocean, processes span many spatial and temporal scales, presenting a key challenge in solving the equations describing them. Figure 1.1, taken from [2], shows the characteristic length and time scales of many atmospheric processes. [2] also contains temporal and spatial spectra of atmospheric motions (which originally come from [3] and [4]), which also show the vast span of scales spread by atmospheric processes. This goes from turbulence, which can have scales of a few metres in space and seconds in time, to planetary waves stretching thousands of kilometres and lasting for days.

One approach to dealing with this non-linearity is to linearise the equations so that the scales do not interact. However for many applications this will simply not produce results that are accurate enough. In the absence of analytic solutions to these fluid equations, we seek to approximate the solutions numerically. This involves making the continuous equations discrete in some way. Typically the continuous time derivative is dealt

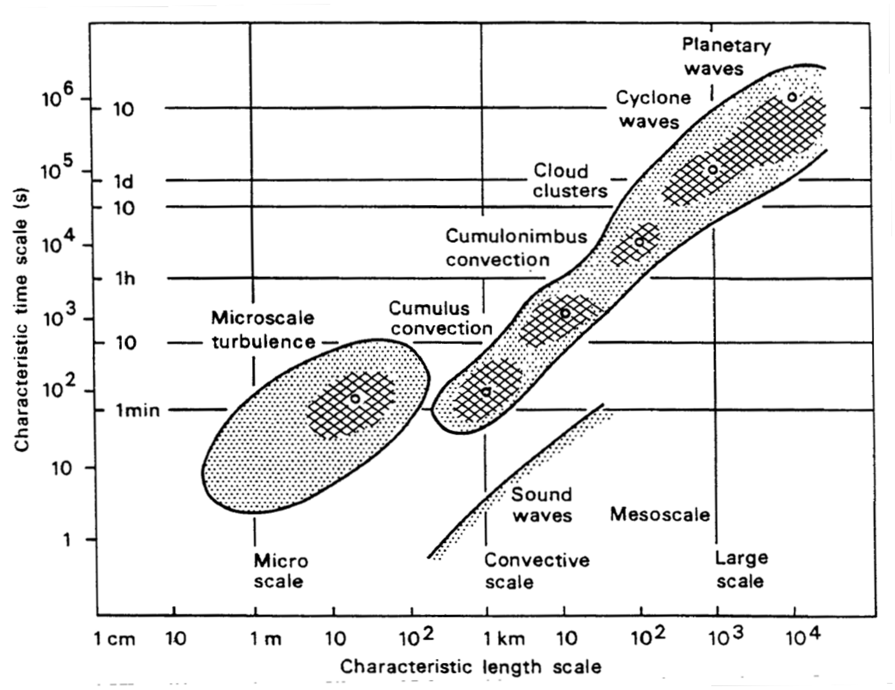


Figure 1.1: A diagram taken from [2] illustrating the characteristic length and time scales of the motion of the atmosphere. Processes span a vast range of both temporal and spatial scales, meaning that in the discretisations used for numerical weather prediction or climate modelling, some motions are inevitably not resolved. The feedback of these unresolved motions upon those resolved by the discretisation necessitate a description of the effects of the unresolved motions within the model.

with by setting the model to take discrete time steps. However, several methods of discretisation in space are common in geophysical fluid applications. In finite difference schemes, fields are stored as a grid of points and spatial derivatives are approximated by differences between the field values. Finite volume schemes divide the domain into cells and write the fluid equations in flux form. Particle methods divide the fluid up into a series of discrete parcels. For finite element methods, the focus of this thesis, space is split into a number of elements, in which the fields are approximated as the sum of a finite series of functions. The equations are typically written in weak form – the differential equations are multiplied by a test function before they are integrated. Such finite element methods will be described in Chapter 2.

In all of these discretisations, there are spatial and temporal scales below which motions cannot be resolved by the numerical model. Limits on the amount of available computational resources inevitably mean that there will be motions below the resolved scales of atmospheric and oceanic models. In global ocean models, this may mean that eddies are unresolved, whilst convection is below the scales resolved by global atmospheric models that are used for numerical weather prediction. Since these unresolved motions interact with those that are resolved, models will not accurately describe the evolution of the resolved flow if the unresolved processes are simply omitted. As described by [2], it is common therefore to attempt to parametrise the effect of the unresolved processes on the resolved ones.

To demonstrate this concept, let us consider an example of the advection of some tracer q by a velocity $\boldsymbol{v}$, so that the evolution of q is given by

$$\frac{\partial q}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} q = 0. \tag{1.1}$$

If q and $\boldsymbol{v}$ can be decomposed into resolved, mean parts $\overline{q}$ and $\overline{\boldsymbol{v}}$, and unresolved, fluctuating parts q' and $\boldsymbol{v}'$, then we can construct an equation for the evolution of $\overline{q}$:

$$\frac{\partial \overline{q}}{\partial t} + \overline{\boldsymbol{v}} \cdot \boldsymbol{\nabla} \overline{q} = -\overline{\boldsymbol{v}' \cdot \boldsymbol{\nabla} q'}, \tag{1.2}$$

where the bar $\bar{\cdot}$ denotes taking some spatial and temporal mean such that $\overline{q'} = 0$ and $\overline{\boldsymbol{v}'} = 0$. The scientific question is then how to parametrise $\overline{\boldsymbol{v}' \cdot \boldsymbol{\nabla} q'}$ in terms of the mean quantities. Similar terms appear in the equation describing the evolution of velocity $\boldsymbol{v}$. Typically this would be $\overline{v_i' v_j'}$, known as the Reynolds stress, with the subscripts i and j denoting the components of the velocity. When parametrising these terms, a common approach is to write them as a diffusion term, such as $K\nabla^2\overline{q}$, with diffusivity K, although there are many more elaborate schemes for parametrising this term. Examples include the Gent and McWilliams parametrisation of [5] and the K-profile parametrisation of [6]. A detailed discussion of this example can be found in [7].

Although such unresolved motions are important, they are not the only physical processes to be parametrised in geophysical models. In models of the atmosphere for numerical weather prediction, other parametrisations include describing transitions between different phases of moisture, radiation and drag due to the Earth's surface. Some of these are illustrated in Figure 1.2. The weather forecasting community often refers to this collection of parametrisations as 'physics', in contrast to the description of the resolved flow which is called 'dynamics'. These parts have typically been separated in operational weather and climate models, and historically there was not much research into how the two parts were coupled together. However, over recent years interest has risen in 'physics-dynamics coupling', which provides the motivation for the research in Part I of this thesis. Another topic of recent interest within this field is stochastic parametrisations, in which the unresolved processes are given a random component to try to capture the uncertainty in these unresolved motions. This is the focus of Part II.

At the same time, the use of finite element methods for modelling geophysical fluids has been growing. In particular, the work of [8] showed that a compatible finite element discretisation (explained in Chapter 2) can possess many properties that are desirable for horizontal discretisations of dynamical cores – the component of the weather or climate model that solves the equations for the resolved flow. Such properties are presented in

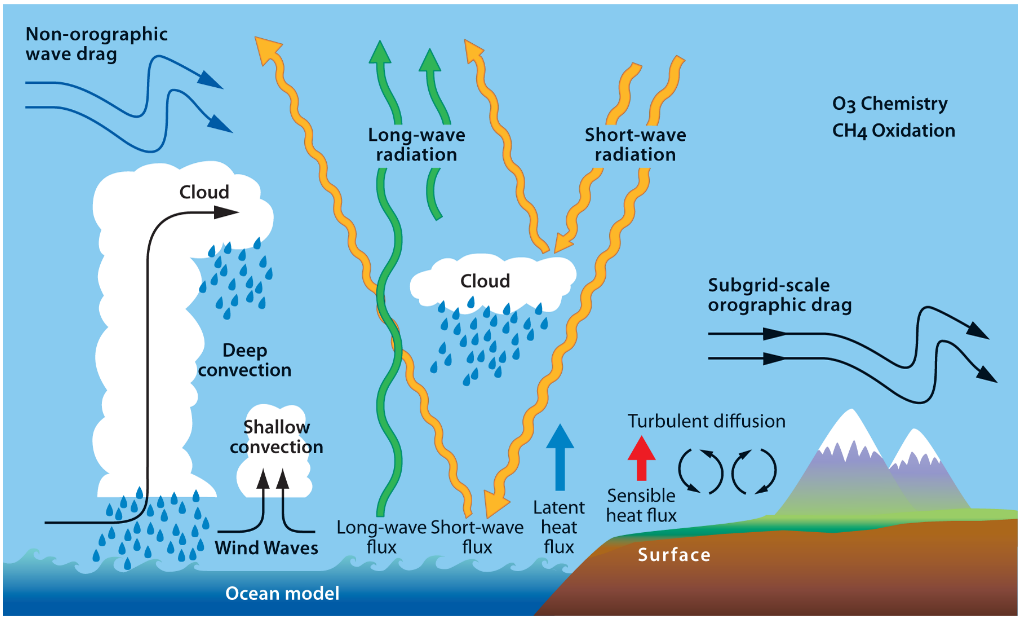


Figure 1.2: Some of the parametrised processes used in numerical weather prediction models. This illustrates atmospheric processes that are not resolved or described by the adiabatic equations solved by the dynamical core, that still need describing to accurately simulate the weather. Image courtesy of ECMWF, the European Centre for Mid-Range Weather Forecasting.

[9]. That these compatible finite element discretisations possess these properties was a critical factor in the choice of the UK Meteorological Office to adopt a compatible finite element discretisation for its next-generation dynamical core [10]. This will be discussed in Chapter 3, which motivates the research of Part I.

1.1 Summary of the Thesis

This thesis is a collection of works under the umbrella theme of coupling resolved and unresolved processes within the context of finite element discretisations of geophysical fluids. In the next chapter we provide an introduction to finite element methods which will be relevant for the rest of the work. The thesis is then split into two parts, with Part I focusing on a compatible finite element discretisation of the atmosphere for numerical weather prediction. Part II concerns a new class of stochastic fluid equations derived in [11] from a variational framework. In this framework, advection has a stochastic component such that circulation is still preserved.

The new research presented as part of this thesis is divided into four chapters, which can be summarised as follows:

- Chapter 4 – a new transport scheme is presented which has second-order accuracy for the lowest-order compatible finite element;

- Chapter 5 – a compatible finite element discretisation of the moist compressible Euler equations is presented;
- Chapter 7 – the statistical properties of a finite element discretisation of the stochastic quasi-geostrophic equation are investigated;
- Chapter 8 – a finite element discretisation of the stochastic Camassa-Holm equation is presented and the formation of peakons within this equation is explored.

1.2 Dissemination

The research presented within this thesis has also been disseminated as journal publications, conference presentations and through open-source software. The publications also presenting this work are:

- Bendall and Cotter (2018), *Statistical properties of an enstrophy conserving finite element discretisation for the stochastic quasi-geostrophic equation* [12], published in Geophysical and Astrophysical Fluid Dynamics;
- Bendall, Cotter and Shipton (2019), *The 'recovered space' advection scheme for lowest-order compatible finite element methods* [13], published in the Journal of Computational Physics;
- Bendall, Gibson, Cotter, Shipton and Shipway (2019), *A compatible finite element discretisation for the moist compressible Euler equations* [14], which appears on the arXiv, and as of writing has been submitted to the Quarterly Journal of the Royal Meteorological Society for review;
- Bendall, Cotter and Holm (2019), *Perspectives on the formation of peakons in the stochastic Camassa-Holm equation* [15], which appears on the arXiv, and as of writing has been submitted to Proceedings of Royal Society A for review.

Conferences and meetings at which this research has been presented include:

- University of Reading SIAM Chapter annual conference, June 2017, *A crash course in wet thermodynamics* (talk);
- Imperial College London Fluids CDT symposium, July 2017, *Discretising the stochastic quasi-geostrophic equations* (poster);
- GungHo network meeting, Exeter, June 2017, *Introducing water to the atmosphere: a 'condensed' summary* (talk);

- UK Fluids Network Meeting, March 2018, *Compatible finite element methods: a new discretisation for weather models* (talk);

- MPE CDT Jamboree, March 2018, *Adding moisture to a finite element model of the atmosphere* (poster);

- Firedrake '18, Imperial College London, June 2018, *Coupling physics to finite element dynamics: the 'recovered' advection scheme* (talk);

- SIAM MPE, Philadelphia, September 2018, *A new discretisation for the stochastic quasi-geostrophic equations* (talk);

- PDEs on the sphere 2019, Montréal, May 2019, *Moisture-dynamics coupling for a compatible finite element dynamical core* (talk).

2 Finite Element Methods

This chapter provides some preliminary material for this thesis, starting with an overview of finite element methods in Section 2.1, which readers already familiar with finite element methods may wish to skip. Two common finite element transport schemes are described in Section 2.2. Section 2.3 provides some background to finite element exterior calculus, which is used to explain compatible finite element methods, which are introduced in Section 2.4. This chapter finishes in Section 2.5 with a brief discussion of our implementation of finite element methods using the Firedrake software.

2.1 An Introduction to the Finite Element

As described by [16], the finite element method evolved from two separate branches of science: the description of stresses and stiffness in structural engineering, and mathematical studies of approximations to partial differential equations. The name 'finite element method' was coined by R. W. Clough, who was an early pioneer in the field [17]. Although we give only outline of finite element methods here, thorough treatments can be found in [18] and [19].

Finite element methods form a major class of Galerkin methods, which, in contrast to finite difference or finite volume discretisations, involve the explicit approximation of fields in finite-dimensional function spaces (rather than simply at a finite number of points or in a finite number of cells). A crucial feature of finite element methods is that equations are written in weak form, which involves multiplying by some test function before integrating, and often manipulating using integration by parts. This has historically often been known as a variational formulation, as in many early problems the solution minimised an error under some energy norm. Fields are then expanded as the sum of a finite number of basis functions. The approximate solution to the partial differential equation is then found by solving a matrix-vector system for the basis coefficients. What distinguishes finite element methods from other Galerkin methods is that the domain Ω is divided into cells, with the basis functions restricted to only their own elements or neighbourhood of elements. An illustration of the division into cells can be found in Figure 2.1. Formally, a *finite element*, as defined by Ciarlet [20], is the triple $(K, P, \mathcal{N})$, where

1. $K \subset \mathbb{R}^n$ is the element domain;

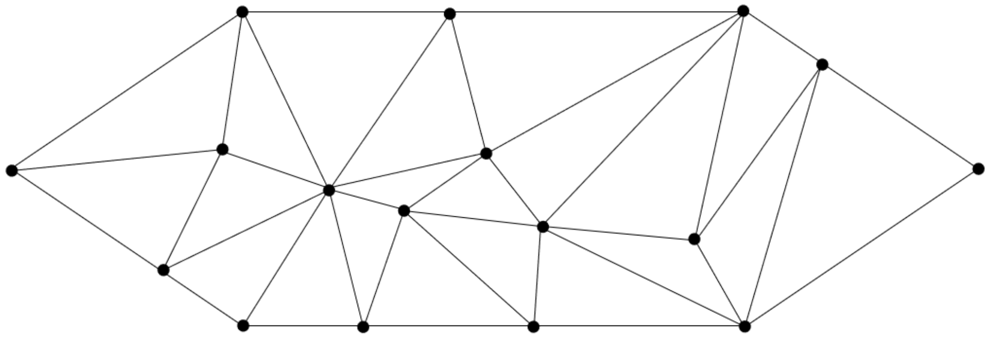


Figure 2.1: A division of a domain into elements. Finite element methods can be easily used on non-uniform meshes.

2. P is a finite dimensional space of functions that are continuous on K, usually polynomial functions;
3. $\mathcal{N} = (n_1, \dots, n_{\dim P})$ is a basis for the dual space P'. The elements of $\mathcal{N}$ are called degrees of freedom (DOFs) or nodes.

The basis $(\psi_1, \dots, \psi_{\dim P})$ for P is often chosen so that each function vanishes at all but one node, so that $n_i(\psi_j) = \delta_{ij}$. A *finite element space* is given by the choice of finite element $(K, P, \mathcal{N})$, along with a division of the domain Ω into cells. The finite element space can be typically characterised by how the function is expanded within a cell and the continuity of its functions between adjacent elements. The properties of the particular finite element method come from the choices of weak formulation and the finite element spaces used.

Part of the popularity of finite element methods is because they provide an application of rigorous functional analysis, as the finite element spaces are finite-dimensional subspaces of Sobolev spaces. In functional analysis, these Sobolev spaces are used to classify the solutions of partial differential equations and their smoothness. Thus the solution to a finite element method may lie in a finite-dimensional subspace of the Sobolev space in which the exact solution to the continuous differential equation lies. In this case the method is called *conforming*. Another significant advantage of these methods is the flexibility with which they can be applied to arbitrary meshes, which can be extremely complicated to use with, for instance, finite difference methods. This advantage has long been exploited by structural and mechanical engineering communities, whose domains such as buildings or cars can be very complex. The compatibility of finite element methods with general grids makes them an obvious choice for adaptive mesh refinement, which is a promising area of interest within discretisation of geophysical fluids.

Some of the most frequently used finite element spaces use the Lagrange polynomials as basis functions. The *continuous Galerkin* spaces use the Lagrange polynomials, and

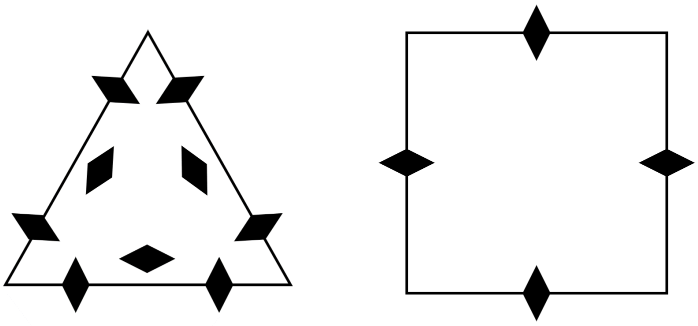


Figure 2.2: Many finite elements are not simply expansions with Lagrange polynomials. (Left) the Brezzi-Douglas-Fortin-Marini space BDFM_1 on a triangular element, which describes a vector with continuous normal and tangential components between cells. (Right) the Raviart-Thomas element RTc_0^f on a quadrilateral cell. It describes vector fields with continuous normal components between elements.

are continuous between cells. When the polynomials are r-th order, we will denote this space by CG_r, although P_r and Q_r are common names for this space on triangular and quadrilateral elements respectively. In contrast, the *discontinuous Galerkin* spaces, written as DG_r, use r-th order polynomials but have no continuity between cells. These are often written for triangular and quadrilateral elements as dP_r and dQ_r. Spaces with mixed polynomial order or continuity requirements are also possible by combining elements through a tensor product structure, which is described in [21] and [22]. We will denote the tensor product of two spaces U and V by $U \otimes V$. Examples of other finite element spaces are the Raviart-Thomas (RT) spaces [23] and the Brezzi-Douglas-Fortin-Marini (BDFM) spaces [24], which are illustrated in Figure 2.2.

2.1.1 An Example for the Helmholtz Equation

As an example demonstrating these concepts, we consider a conforming finite element discretisation of the Helmholtz equation, which often appears in geophysical fluid models as an equation for the pressure. Let the domain Ω be the unit square, divided into a uniform grid of quadrilateral elements. The equation that we will attempt to solve for variable q is

$$\left(k^2 - \nabla^2\right) q = F(x, y), \tag{2.1}$$

subject to the boundary condition that $\hat{\boldsymbol{n}} \cdot \boldsymbol{\nabla} q = 0$ on the boundary of the domain $\partial\Omega$, where $\hat{\boldsymbol{n}}$ is the unit normal to the edge of the domain. We take the constant $k = 100$ and the forcing F as

$$F = \cos(2\pi x)\cos(2\pi y). \tag{2.2}$$

This can be cast in weak form by multiplying by a test function ψ and integrating. Under assumptions of continuity of q between cells, we integrate the Laplacian term by parts

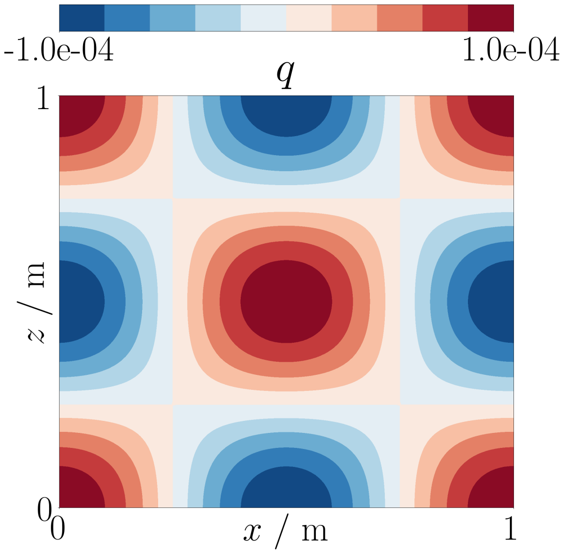


Figure 2.3: The solution q of the Helmholtz equation (2.1), which often appears in models of geophysical fluids. This solution was obtained using the finite element formulation (2.3) with quadrilateral elements of size $\Delta x = \Delta y = 0.01$. This is used as a benchmark for the solution shown in Figure 2.4, which used the compatible finite element formulation outlined in Section 2.4.

and use the boundary condition $\widehat{\mathbf{n}} \cdot \boldsymbol{\nabla} q = 0$ on $\partial\Omega$ to obtain

$$\int_\Omega k^2 \psi q \, dx + \int_\Omega \boldsymbol{\nabla}\psi \cdot \boldsymbol{\nabla} q \, dx = \int_\Omega \psi F \, dx. \tag{2.3}$$

The Galerkin approximation that we make is that the function q and the test function ψ are in Q_1, the space of piecewise, continuous linear functions. The second derivative of q is zero everywhere in this finite element space, which motivates our integration by parts of the Laplacian term. The solution to the problem is the q that satisfies (2.3) for all $\psi \in Q_1$. The test function ψ and the function q are then expanded using the basis functions of Q_1, turning this into a matrix-vector problem to solve for the coefficients of q in this expansion. The solution is shown in Figure 2.3.

2.2 Two Common Finite Element Transport Schemes

This section serves as a summary of two common finite element methods for describing the transport of a tracer q by a known velocity $\boldsymbol{v}$, obeying the "advective" form of the transport equation

$$\frac{\partial q}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} q = 0, \tag{2.4}$$

where $\widehat{\mathbf{n}} \cdot \boldsymbol{v} = 0$ on the boundary $\partial\Omega$ of the domain Ω, so that there is no inflow or outflow. The velocity will lie in the vector Q_1 space. Both schemes described here will be relevant later in the thesis, and begin with multiplying by some test function ψ which will be in the same space as q to obtain a square system to solve. The approaches differ in the assumptions about the continuity of q, but we also use a different time stepping scheme with each to illustrate two different choices of time discretisation.

2.2.1 Conforming Implicit Midpoint Transport

First, we will assume continuous q and ψ. Integrating and manipulating using integration by parts gives

$$\int_\Omega \psi \frac{\partial q}{\partial t}\, dx - \int_\Omega q \boldsymbol{\nabla} \cdot (\psi \boldsymbol{v})\, dx = 0. \tag{2.5}$$

For the time discretisation, we are attempting to solve for $q^{(n+1)}$ at the $(n+1)$-th time step given $q^{(n)}$, with time step Δt. In this example we use an *implicit midpoint* scheme, which involves replacing every q with $q^{(n+1/2)} = \frac{1}{2}\left(q^{(n)} + q^{(n+1)}\right)$, except for the time derivative which is simply $\left(q^{(n+1)} - q^{(n)}\right)/\Delta t$. Advantages of this time stepping scheme are that it is unconditionally stable [25] and it preserves quadratic integral quantities that are invariant in the continuous equations [26]. In our case this means that $\int_\Omega q^2\, dx$ is preserved when $\boldsymbol{\nabla} \cdot \boldsymbol{v} = 0$. Although choosing a conforming spatial discretisation prevents the need for discretising terms on interior facets (unlike the method we will see in Section 2.2.2), a major disadvantage to conforming discretisations in general is that they require the solution of problems globally over the domain, with basis functions overlapping from one element to another. This can make it difficult to solve efficiently.

The final equation can be written as

$$\int_\Omega \psi q^{(n+1)}\, dx = \int_\Omega \psi q^{(n)}\, dx + \Delta t \int_\Omega q^{(n+1/2)} \boldsymbol{\nabla} \cdot (\psi \boldsymbol{v})\, dx = 0. \tag{2.6}$$

2.2.2 Discontinuous Galerkin Upwinding

In this discretisation, the assumptions about continuity of q and test function ψ are relaxed. The weak form obtained is then

$$\int_\Omega \psi \frac{\partial q}{\partial t}\, dx + \int_\Omega \boldsymbol{\nabla} \cdot (\psi q \boldsymbol{v})\, dx - \int_\Omega q \boldsymbol{\nabla} \cdot (\psi \boldsymbol{v})\, dx = 0. \tag{2.7}$$

The divergence theorem gives

$$\int_\Omega \psi \frac{\partial q}{\partial t}\, dx + \sum_K \int_{\Gamma_K} \psi q \boldsymbol{v} \cdot \widehat{\boldsymbol{n}}\, dS + \int_{\partial\Omega} \psi q \boldsymbol{v} \cdot \widehat{\boldsymbol{n}}\, ds - \int_\Omega q \boldsymbol{\nabla} \cdot (\psi \boldsymbol{v})\, dx = 0, \tag{2.8}$$

where Γ_K represents the interior facets of the element K. The surface measure on interior facets is written dS, with ds representing the surface measure on exterior facets. On facets, $\widehat{\boldsymbol{n}}$ is always the outward pointing normal vector and interior facets are summed over twice, with $\widehat{\boldsymbol{n}}$ pointing in the opposite direction in each case. The third integral in equation (2.8) will vanish due to to our boundary conditions.

We will choose q and ψ to be in the discontinuous space dQ_1. In this case, the interior facet integral in (2.8) does not vanish, and the challenge is how to discretise it. A discussion of some of the ways of discretising these fluxes in this context can be found in [27]. Our choice is the *upwind flux*, in which the equation becomes

$$\int_\Omega \psi \frac{\partial q}{\partial t}\, dx + \int_\Gamma [\![\psi]\!]_+ \, q^\dagger \left(\boldsymbol{\nu}\cdot\widehat{\boldsymbol{n}}^+\right)\, dS - \int_\Omega q\boldsymbol{\nabla}\cdot\,(\psi\boldsymbol{\nu})\, dx = 0, \tag{2.9}$$

where $\Gamma = \cup_K \Gamma_K$. Each side of an interior facet is labelled arbitrarily with $+$ or $-$, and the double square brackets represent the jump in values on either side of the facet

$$[\![\psi]\!]_+ = \psi^+ - \psi^-. \tag{2.10}$$

The upwind value of q is represented by $q^\dagger$, which assumes the value

$$q^\dagger = \begin{cases} q^+ & \text{if } \boldsymbol{\nu}\cdot\widehat{\boldsymbol{n}}^+ > 0, \\ q^- & \text{if } \boldsymbol{\nu}\cdot\widehat{\boldsymbol{n}}^+ \leqslant 0. \end{cases} \tag{2.11}$$

For the time discretisation, such an upwinding scheme is commonly used with a Runge-Kutta scheme, such as is the subject of [27]. The scheme that we will use later in this work is a third-order strong stability-preserving Runge-Kutta scheme (SSPRK-3), see for instance [28] or [27]. In this scheme, we first introduce a simple forward Euler time step, to find the difference in q, which we denote by $\mathcal{L}q^{(n)}$, such that

$$\int_\Omega \psi\mathcal{L}q^{(n)}\, dx = -\Delta t\int_\Gamma [\![\psi]\!]_+ \, q^{(n)\dagger}(\boldsymbol{\nu}\cdot\widehat{\boldsymbol{n}}^+)\, dS + \Delta t\int_\Omega q^{(n)}\boldsymbol{\nabla}\cdot\,(\psi\boldsymbol{\nu})\, dx. \tag{2.12}$$

The solution at the $(n+1)$-th time step is then found by taking the three steps

$$q^{(1)} := q^{(n)} + \mathcal{L}q^{(n)}, \tag{2.13a}$$

$$q^{(2)} := \frac{3}{4}q^{(n)} + \frac{1}{4}q^{(1)} + \frac{1}{4}\mathcal{L}q^{(1)}, \tag{2.13b}$$

$$q^{(n+1)} := \frac{1}{3}q^{(n)} + \frac{2}{3}q^{(2)} + \frac{2}{3}\mathcal{L}q^{(2)}. \tag{2.13c}$$

Some of the main appeals of discontinuous solutions are that they can better represent shocks and steep gradients in solutions (such as weather fronts) and they do not require the solution of a global matrix-vector equation as the system is block-diagonal, making the problem simpler to solve efficiently on parallel computers. However one of the major drawbacks of this method is that it is only conditionally stable. The instability is determined by the Courant number $c = |\boldsymbol{\nu}|\Delta t/\Delta x$, for cell width Δx. This instability occurs when c breaches some critical value, given by the Courant-Friedrichs-Lewy condition (CFL). For instance, [27] quotes the CFL limit for the scheme described in this section as 0.409.

Although we could have used an implicit midpoint scheme to integrate in time, such Runge-Kutta schemes allow easy application of *slope-limiters*. These are used as the discretisation does not preserve the property of the continuous equation that the solution remains bounded by its initial maximum and minimum values. Limiters can prevent overshoots and undershoots generating unphysical values, such as negative concentrations of moisture.

2.3 Finite Element Exterior Calculus

The goal of this section is to explain the discrete de Rham complex, which is the mathematical structure underlying the compatible finite element methods which will be described in Section 2.4. First, some concepts from exterior calculus must be introduced.

2.3.1 Exterior Calculus

Differential k*-forms* are the basic building block of exterior calculus, and are mathematical objects that take k vector fields and return a real number. They are examples of algebraic k-forms, which are alternating linear maps from k vectors to the real numbers. In the case of differential k-forms, the mapping is from the tangent spaces of a smooth manifold. Differential k-forms are the integrands of line, surface and volume integrals, or of those integrals over higher-dimensional manifolds. Circulation is an example of a 1-form in fluid dynamics, whilst a mass flux through a surface is a 2-form.

More formally, a differential k-form can be described given an n-dimensional smooth manifold Ω, with tangent space at point x given by $T_x\Omega$. For vector fields $v_1, \dots, v_k \in T_x\Omega$, a differential form is an alternating linear map $\omega(v_1, \dots, v_k)$ assigning an element ω_x to each $x \in \Omega$. Let the space of differential k-forms on Ω be $\Lambda^k(\Omega)$. Note that there are no differential k-forms for $k > n$. For a visual introduction to differential forms, see [29], or for complete explanation in the context of finite element methods see [30].

Differential k-forms are equipped with an *exterior derivative*, denoted by d, which maps from $\Lambda^k(\Omega)$ to $\Lambda^{k+1}(\Omega)$. Following [30], for vector fields $v_1, \dots, v_{k+1} \in T_x\Omega$, the exterior derivative of $\omega \in \Lambda^k(\Omega)$ at x can be defined by

$$d\omega_x(v_1, \dots, v_{k+1}) = \sum_{j=1}^{k+1} (-1)^{j+1} \partial_{v_j} \omega_x(v_1, \dots, \widehat{v}_j, \dots v_{k+1}), \tag{2.14}$$

with the hat $\widehat{\cdot}$ denoting a suppressed argument. An important property of the exterior derivative is that the second derivative $d^2 = 0$.

A second operation on differential k-forms that we will briefly mention is the *interior product*, denoted by $\lrcorner$, which takes a vector field ν and $\omega \in \Lambda^k$ and returns a differential $(k-1)$-form. As described in [29], this can be written

$$\nu \lrcorner\, \omega(\nu_1, \dots, \nu_k) = \omega(\nu, \nu_1, \dots \nu_{k-1}). \tag{2.15}$$

Relevant to us are the two- or three-dimensional physical spaces $\mathbb{R}^2$ and $\mathbb{R}^3$. In these situations the exterior derivative can correspond to well-known vector calculus derivatives, grad, div and curl. The exterior derivative of the coordinate function x_i is the 1-form dx_i. The basic 1-form is the integrand $\boldsymbol{\omega} \cdot d\mathbf{x}$ to line integrals.

The *de Rham complex* is the sequence of spaces and mappings

$$0 \to \Lambda^0(\Omega) \xrightarrow{d} \Lambda^1(\Omega) \xrightarrow{d} \dots \xrightarrow{d} \Lambda^n(\Omega) \to 0, \tag{2.16}$$

where the spaces are the spaces of differential k-forms and the mappings are given by the exterior derivative. This is an exact complex, with no space being mapped to $\Lambda^0(\Omega)$ and nothing being mapped from $\Lambda^k(\Omega)$ by d.

There are also subspaces of $\Lambda^k(\Omega)$ containing differential k-forms with particular properties, which may also form their own complexes. Important to us are spaces of differential k-forms that also lie within particular Sobolev spaces, that categorise their smoothness.* The two complexes that we will consider are L^2 de Rham complexes when the domains are $\mathbb{R}^2$ and $\mathbb{R}^3$. These complexes are

$$0 \to H^1(\mathbb{R}^2) \xrightarrow{\boldsymbol{\nabla}^\perp} H(\text{div}; \mathbb{R}^2) \xrightarrow{\boldsymbol{\nabla}\cdot} L^2(\mathbb{R}^2) \to 0, \tag{2.17}$$

$$0 \to H^1(\mathbb{R}^3) \xrightarrow{\boldsymbol{\nabla}} H(\text{curl}; \mathbb{R}^3) \xrightarrow{\boldsymbol{\nabla}\times} H(\text{div}; \mathbb{R}^3) \xrightarrow{\boldsymbol{\nabla}\cdot} L^2(\mathbb{R}^3) \to 0. \tag{2.18}$$

Here, L^2 is the space of all square-integrable functions and the Sobolev space H^1 is the space of all functions which are square-integrable and have a weak derivative that is square-integrable. Similarly, $H(\text{div})$ is the space of all functions which are both square-integrable and have a weak divergence that is square-integrable, whilst $H(\text{curl})$ is the space of all functions that are square-integrable and also have a square-integrable weak curl. The $\boldsymbol{\nabla}^\perp$ operator is perpendicular to the gradient $\boldsymbol{\nabla}$, and is defined in Cartesian coordinates in $\mathbb{R}^2$ to be

$$\boldsymbol{\nabla}^\perp = \begin{pmatrix} -\partial_y \\ \partial_x \end{pmatrix}. \tag{2.19}$$

*Although we avoid defining Sobolev spaces here, they are defined in [30]. For a rigorous guide to functional analysis and Sobolev spaces, the reader could see [31], for example.

2.3.2 Finite Element Exterior Calculus

Finite-dimensional spaces of differential k-forms are applicable to finite element discretisations. These can also be combined to form complexes connected by analogues of the exterior derivative. For instance, let us consider $\mathcal{P}_r\Lambda^k(\mathbb{R}^n)$, the space of polynomial differential k-forms of degree r, upon $\mathbb{R}^n$. This space forms the polynomial complex

$$0 \to \mathcal{P}_r\Lambda^0(\mathbb{R}^n) \overset{d}{\to} \mathcal{P}_{r-1}\Lambda^1(\mathbb{R}^n) \overset{d}{\to} \dots \overset{d}{\to} \mathcal{P}_{r-n}\Lambda^n(\mathbb{R}^n) \to 0. \tag{2.20}$$

For $n = 3$, an example 1-form with $r = 2$ might look like

$$yz\,dx + xz\,dy + xy\,dz. \tag{2.21}$$

The final step in using such differential k-forms for finite element methods is to consider these forms restricted to the shape of cell that composes the mesh. The most well-known of these cases are documented in the periodic table of finite elements [32].

A de Rham complex used for finite element methods is then a family of finite element differential forms equipped with exterior derivatives. The main family that will be used in this thesis is $\mathcal{Q}_r^-\Lambda^k(\Omega)$, which describes polynomial forms on quadrilateral or cubic cells, which forms a column in the periodic table [32]. The complexes that are relevant to us are

$$0 \to Q_{r+1}(\mathbb{R}^1) \xrightarrow{d/dx} dQ_r(\mathbb{R}^1) \to 0, \tag{2.22a}$$

$$0 \to Q_{r+1}(\mathbb{R}^2) \xrightarrow{\nabla^\perp} RTc_r^f(\mathbb{R}^2) \xrightarrow{\nabla\cdot} dQ_r(\mathbb{R}^2) \to 0, \tag{2.22b}$$

$$0 \to Q_{r+1}(\mathbb{R}^3) \xrightarrow{\nabla} Nc_r^e(\mathbb{R}^3) \xrightarrow{\nabla\times} Nc_r^f(\mathbb{R}^3) \xrightarrow{\nabla\cdot} dQ_r(\mathbb{R}^3) \to 0, \tag{2.22c}$$

where r denotes the degree of the element, which typically gives its approximation order[†]. As previously mentioned, Q_r and dQ_r represent elements of scalar fields expanded in polynomials of degree r on quadrilateral cells that are respectively continuous and discontinuous between cells. The Raviart-Thomas element RTc_r^f describes vector fields with continuous normal components, and corresponds to a discrete $H(\mathrm{div})$ space. The Nédélec elements Nc_r^e and Nc_r^f, sometimes also called Raviart-Thomas-Nédélec elements, are $H(\mathrm{curl})$ and $H(\mathrm{div})$ spaces in three-dimensions. They respectively describe vector fields with continuous tangential and normal components, with e and f indicating nodes associated with edges or facets. For illustration of the spaces used in this thesis for the discretisation of the equations describing the atmosphere, see Table 3.1.

[†] There are two numbering conventions frequently used in the literature. The first (used here) gives the order of the approximation of the element. The other, used for instance in the periodic table of the finite elements [32], gives the degree of the highest order polynomial of the element. Often these are the same, but as an example the Raviart-Thomas elements will differ under this convention.

2.4 Compatible Finite Element Methods

A mixed finite element method solves coupled partial differential equations with more than one variable, where the variables may or may not lie in different finite element spaces. For instance, the Helmholtz example from Section 2.1.1 can be recast as

$$\nabla q = \mathbf{u}, \tag{2.23}$$

$$k^2 q - \nabla \cdot \mathbf{u} = F, \tag{2.24}$$

giving two equations to solve simultaneously for q and $\mathbf{u}$. These methods are known as *compatible* or *mimetic* when the finite element function spaces are taken from a finite element de Rham complex, such that the spaces of the variables correspond to those from the de Rham complex of the continuous partial differential equations that are attempted to be solved. A consequence of this is that the discrete differential operators are analogues of the differential operators relating the respective continuous spaces. The discrete differential operators then satisfy familiar identities from vector calculus, such as

$$\nabla \cdot \nabla \times \mathbf{f} = 0, \quad \nabla \times \nabla q = 0, \tag{2.25}$$

for any scalar field q or vector field $\mathbf{f}$.

Crucially, the motivation for using compatible finite element methods is that they can create stable discretisations, when general mixed formulations may not. As described by [33], the discrete and continuous spaces can be related via diagrams such as the following example in two dimensions:

$$\begin{tikzcd}
H^1(\mathbb{R}^2) \arrow[r, "\nabla^\perp"] \arrow[d, "\Pi^Q"] & H(\mathrm{div};\mathbb{R}^2) \arrow[r, "\nabla\cdot"] \arrow[d, "\Pi^{RT}"] & L^2(\mathbb{R}^2) \arrow[d, "\Pi^{dQ}"] \\
Q_{r+1}(\mathbb{R}^2) \arrow[r, "\nabla^\perp"] & RTc^f_r(\mathbb{R}^2) \arrow[r, "\nabla\cdot"] & dQ_r(\mathbb{R}^2)
\end{tikzcd} \tag{2.26}$$

where Π^Q, Π^{RT} and Π^{dQ} are the projections to the respective finite element spaces. The commutativity of the operations shown in such diagrams is used to show the stability of the corresponding mixed finite element formulations, such as in [8].

2.4.1 Another Example with the Helmholtz Equation

We now return to our Helmholtz example from Section (2.1.1), cast as two coupled first-order PDEs:

$$\nabla q = \mathbf{u}, \tag{2.27a}$$

$$k^2 q - \nabla \cdot \mathbf{u} = F, \tag{2.27b}$$

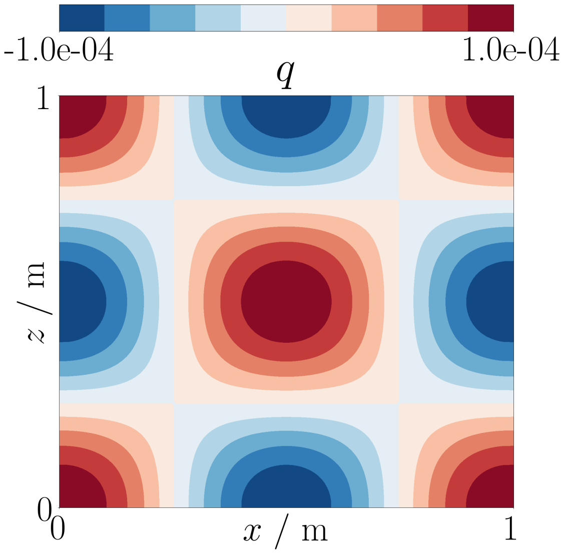


Figure 2.4: The solution q of the Helmholtz equation (2.1) using the compatible finite element formulation (2.28) with quadrilateral elements of size $\Delta x = \Delta y = 0.01$. Using the compatible finite element approach can deliver stable results comparable with that shown in Figure 2.3. This may not be the case for another mixed formulation if the function spaces are not chosen to be compatible.

with the boundary condition $\widehat{\boldsymbol{n}} \cdot \boldsymbol{u} = 0$ on $\partial\Omega$. To obtain the mixed weak formulation, we multiply by test functions $\boldsymbol{\phi}$ and ψ, which are in the same spaces as $\boldsymbol{u}$ and q respectively. Following integration by parts, application of divergence theorem and assumptions about the continuity of the fields, we obtain

$$\int_{\partial\Omega} q\boldsymbol{\phi} \cdot \widehat{\boldsymbol{n}} \, ds - \int_{\Omega} q\boldsymbol{\nabla} \cdot \boldsymbol{\phi} \, dx = \int_{\Omega} \boldsymbol{\phi} \cdot \boldsymbol{u} \, dx, \tag{2.28a}$$

$$\int_{\Omega} k^2 \psi q \, dx - \int_{\Omega} \psi \boldsymbol{\nabla} \cdot \boldsymbol{u} \, dx = \int_{\Omega} \psi F \, dx. \tag{2.28b}$$

To take a *compatible* finite element approach, $\boldsymbol{u}$ and q are chosen to lie in spaces from an appropriate complex, with $\boldsymbol{u} \in V_u \subset H(\Omega; \text{div})$ and $q \in V_q \subset L^2(\Omega)$. The solution is the pair $(\boldsymbol{u}, q)$ that solves (2.28) for all $\boldsymbol{\phi} \in V_u$ and for all $\psi \in V_q$. As before, we use a uniform quadrilateral mesh of the unit square. A natural family of finite element spaces to choose is $Q^{-}\Lambda^{k}$ from the periodic table of finite elements. We will take the lowest order spaces, so that $V_u = \text{RTc}^{f}{}_{0}$ and $V_q = dQ_0$. The solution to this is shown in Figure 2.4.

2.5 Firedrake

We have outlined finite element methods in this section, giving a taste of their mathematical elegance and also their flexibility. However, a major challenge in using finite element methods can be their numerical implementation. There are many components that require programming: triangulations of the domain into elements; schemes for turning the weak formulation into discrete equations; mappings between physical elements and a reference element; quadrature rules for numerical integration; strategies for solving matrix-vector problems. The code should also be efficient, and ideally able to exploit parallel computational architectures.

Fortunately this thesis was completed without the author needing to write the code to explicitly perform these tasks. The Firedrake software [34], which was used for all the numerical results in this thesis, is developed by a team centred at Imperial College London, and automates all of the procedures listed above (and many more). Firedrake originally grew from the FEniCS project [35], another software package providing an abstraction for solving partial differential equations using finite element methods. Both software packages auto-generate low-level code to solve the weak formulations provided by the user in UFL, the *unified form language* [36]. Firedrake and FEniCS then convert the Python-based UFL into low-level code; in the case of Firedrake this is done by the *two-stage form compiler* TSFC [37] and FInAT [38, 39], which create c code that assembles the finite element forms into kernels to be run for each cell in the mesh. A significant difference between Firedrake and FEniCS is the way that these kernels are organised and executed, which in Firedrake is performed by PyOP2 [34]. In the case of Firedrake, the Python language is used for levels of code down to the executed kernels, allowing information to be more easily passed to the low-level code generation. The third party PETSc software package provides preconditioners and numerical solvers for the large linear systems produced by PyOP2. Another component of Firedrake is Slate, which allows use of more advanced solver techniques for solving discontinuous and mixed Galerkin problems known as hybridisation techniques [40]. An overview of the components of Firedrake and how they interact with one another is shown in Figure 2.5. Important features of Firedrake used in this thesis are quadrilateral elements on extruded meshes [41, 42], with tensor product elements [22].

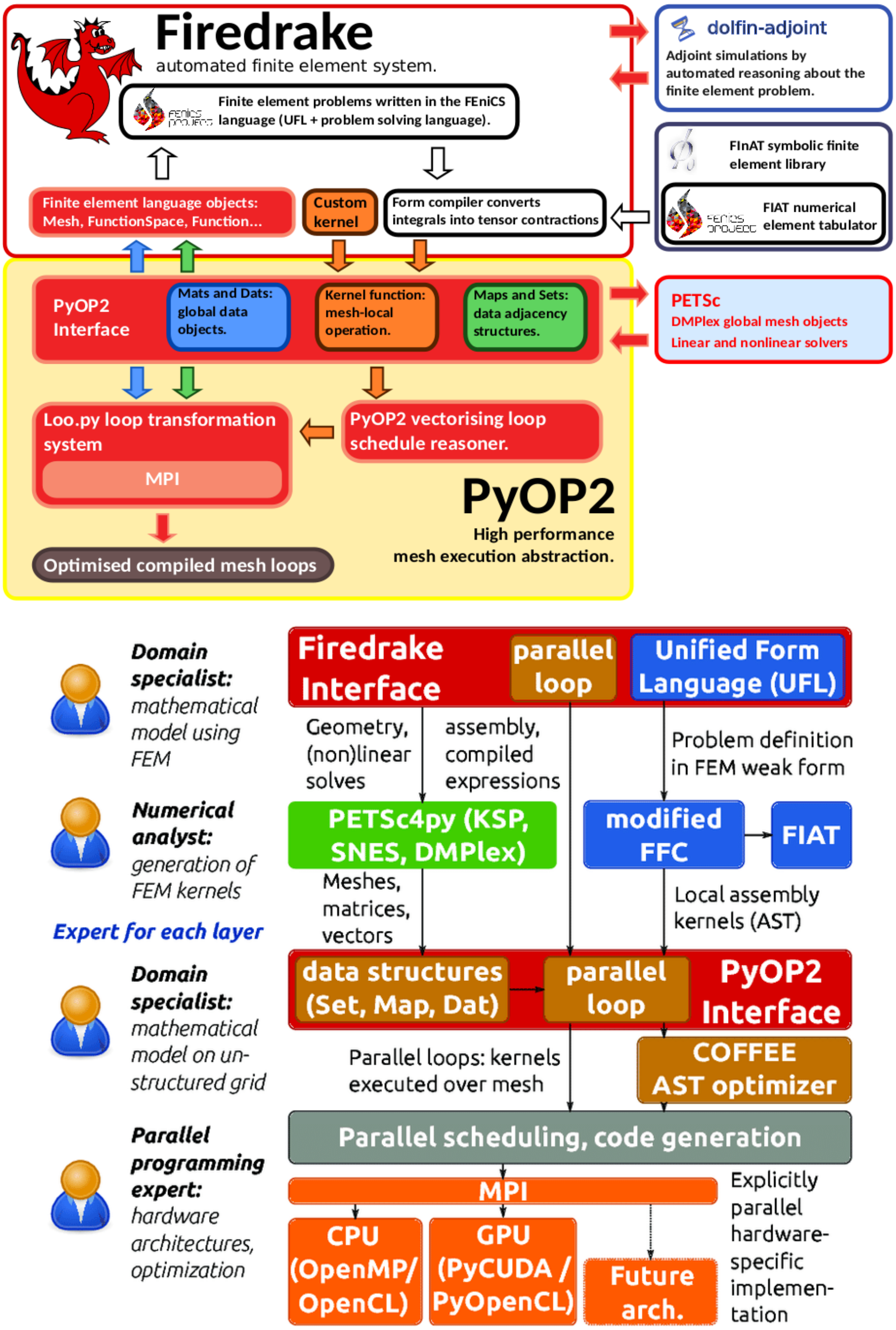

Firedrake
automated finite element system.
Finite element problems written in the FEniCS language (UFL + problem solving language).
Finite element language objects: Mesh, FunctionSpace, Function...
Custom kernel
Form compiler converts integrals into tensor contractions
dolfin-adjoint
Adjoint simulations by automated reasoning about the finite element problem.
FInAT symbolic finite element library
FIAT numerical element tabulator
PETSc
DMPlex global mesh objects
Linear and nonlinear solvers
PyOP2 Interface
Mats and Dats: global data objects.
Kernel function: mesh-local operation.
Maps and Sets: data adjacency structures.
Loo.py loop transformation system
MPI
PyOP2 vectorising loop schedule reasoner.
PyOP2
High performance mesh execution abstraction.
Optimised compiled mesh loops
Domain specialist: mathematical model using FEM
Numerical analyst: generation of FEM kernels
Expert for each layer
Domain specialist: mathematical model on unstructured grid
Parallel programming expert: hardware architectures, optimization
Firedrake Interface
parallel loop
Unified Form Language (UFL)
Geometry, (non)linear solves
assembly, compiled expressions
Problem definition in FEM weak form
PETSc4py (KSP, SNES, DMPlex)
modified FFC
FIAT
Meshes, matrices, vectors
Local assembly kernels (AST)
data structures (Set, Map, Dat)
parallel loop
PyOP2 Interface
Parallel loops: kernels executed over mesh
COFFEE AST optimizer
Parallel scheduling, code generation
MPI
CPU (OpenMP/ OpenCL)
GPU (PyCUDA / PyOpenCL)
Future arch.
Explicitly parallel hardware-specific implementation

Figure 2.5: Schematics of the components of the Firedrake software, which automates code generation for the solution of partial differential equations using finite element methods. These help to illustrate both how complex finite element software can be and how the different components of this software can be abstracted. (Top) a newer graphic, courtesy of David Ham, illustrating how the current components of Firedrake are expressed abstractly and their interaction with one another. (Bottom) an older graphic from [43] that shows how different scientists might interact with different aspects of the software. The work of presented in this thesis was predominantly completed with interaction through the Firedrake interface.

I

Compatible Finite Element Discretisations

3 GungHo and Gusto

For a convergent numerical discretisation of a differential equation, increasing its resolution leads to more accurate solutions. In the context of dynamical cores of atmospheric models, this means using more degrees of freedom (DOFs) or reducing the time step. This, however, relies on more computational resources, which typically now means performing calculations in parallel over increasing numbers of processors. The communication between the processors is then important for determining the efficiency of the numerical method.

At the UK Met Office, the dynamical core for global simulations uses a latitude-longitude grid as a mesh over the sphere. This choice of grid grants many properties important for the accuracy and stability of the discretisation, which are discussed in [9]. However, the convergence of meridians at the poles is leading to a bottleneck in the rate of communication between the processors dealing with this part of the model – and thus also a bottleneck in the scalability of the method [10]. As part of a search for a new horizontal grid, [8] investigated the use of compatible finite element discretisations with the rotating shallow-water equations. The use of this discretisation would facilitate the use of meshes over the sphere that are much more general than just the latitude-longitude grid. Crucially, many of the desirable properties presented in [9] can still be attained by using compatible finite element discretisation.

The GungHo project's aim is to develop a new compatible finite element dynamical core for use in the operational model of the UK Met Office [44]. Within this project sits Gusto: a dynamical core toolkit using the Firedrake finite element software described in Section 2.5. It provides a library for discretising a hierarchy of geophysical fluid equations: the compressible Euler equations, Boussinesq equations and shallow-water equations. These can be simulated with different meshes, different degrees of spaces and with different transport schemes, providing a sandpit for the investigation of the choices involved with building the next-generation dynamical core. Some of these choices involve how the dynamical core will be coupled to the physical parametrisations once the dynamical core is different. Chapter 5 presents research contributing to this area.

In this chapter, we start by giving more detail on the motivation for developing a dynamical core using a compatible finite element method. We then describe the model details of Gusto, which provides the background for the main results of this part of the thesis: the presentation of a new transport scheme in Chapter 4 and the development of a discretisation of the moist compressible Euler equations in Chapter 5. Much of this chapter is adapted from [14].

3.1 The GungHo Project

3.1.1 Equations for a Dry Dynamical Core

The full compressible Euler equations used in a dynamical core to describe the behaviour of a dry atmosphere on a rotating sphere, such as those given in [45] or [46], can be written as

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\boldsymbol{v} + \mathbf{f}\times\boldsymbol{v} + \frac{1}{\rho}\boldsymbol{\nabla}p + \boldsymbol{\nabla}\Phi = \boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\boldsymbol{v}) + \mathbf{F}^{v}, \tag{3.1a}$$

$$\frac{\partial \theta_d}{\partial t} + (\boldsymbol{v}\cdot\boldsymbol{\nabla})\theta_d = \boldsymbol{\nabla}\cdot(K\boldsymbol{\nabla}\theta_d) + F^{\theta}, \tag{3.1b}$$

$$\frac{\partial \rho_d}{\partial t} + \boldsymbol{\nabla}\cdot(\rho_d\boldsymbol{v}) = 0, \tag{3.1c}$$

where $\boldsymbol{v}$ is the air velocity, $\mathbf{f} = f\widehat{\mathbf{k}}$ represents the Coriolis parameter multiplied by the vertical unit vector, ρ_d is the dry air density, p is the air pressure and Φ is the gravitational potential, such that $\boldsymbol{\nabla}\Phi = \widehat{\mathbf{k}}g$, with g the acceleration due to gravity. The density and pressure are related via the ideal gas law for dry air,

$$p = \rho_d R_d T, \tag{3.2}$$

for temperature T and specific gas constant for dry air, R_d. These variables also define the dry potential temperature θ_d via

$$\theta_d := T\left(\frac{p_R}{p}\right)^{\frac{R_d}{c_{pd}}}, \tag{3.3}$$

where p_R is a reference pressure (usually 1×10^5 Pa) and c_{pd} is the specific heat capacity of dry air at constant pressure. $\mathbf{F}^{v}$ and F^{θ} describe sources and sinks of velocity and potential temperature respectively. These terms describe the 'physics': the non-adiabatic, dissipative or unresolved processes. We explicitly also include two such terms, describing an artificial diffusion in the equations for $\boldsymbol{v}$ and θ_d, where K is the diffusivity. These artificial diffusion terms will be treated like physics parametrisations in our model which motivates the use of the name *compressible Euler equations* as opposed to the Navier-Stokes equations. In many test-cases these terms will be zero. All variables are expressed in a frame of motion that rotates with the sphere so that the Earth's surface appears stationary, which leads to the appearance of the Coriolis force in (3.1a).

Another common form of (3.1a) involves rewriting the pressure gradient force to give

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{v} \cdot \boldsymbol{\nabla})\boldsymbol{v} + \mathbf{f} \times \boldsymbol{v} + c_{pd}\theta_d \boldsymbol{\nabla}\Pi + \boldsymbol{\nabla}\Phi = \boldsymbol{\nabla} \cdot (K\boldsymbol{\nabla}\boldsymbol{v}) + \mathbf{F}^{\nu}. \tag{3.4}$$

Here Π is the Exner pressure, defined to be

$$\Pi := \left(\frac{p}{p_R}\right)^{\frac{R_d}{c_{pd}}}, \tag{3.5}$$

which gives $\theta_d = T/\Pi$. The advective derivative of (3.4) can also be rewritten to cast the momentum equation in vector-invariant form

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{\nabla} \times \boldsymbol{v}) \times \boldsymbol{v} + \mathbf{f} \times \boldsymbol{v} + \frac{1}{2}\boldsymbol{\nabla}v^2 + c_{pd}\theta_d \boldsymbol{\nabla}\Pi = \boldsymbol{\nabla} \cdot (K\boldsymbol{\nabla}\boldsymbol{v}) + \mathbf{F}^{\nu}. \tag{3.6}$$

These equations describe a dry atmosphere; the presence of moisture leads to a different equation of state, while the latent heat release associated with condensation changes the thermodynamic equation. The equations describing a wet atmosphere can be found in Section 5.1.

3.1.2 Desired Properties of a Dynamical Core

When building a numerical model for weather or climate simulation, there is a trade-off between the accuracy of the solution and the time taken to run, constrained by the amount of available computational resources. The goal is therefore to design a model that is as accurate as possible, given an acceptable run time. However, these models are made up of several large components, of which the physics and dynamics are just two. Examples of other components might include data assimilation, an ocean model or a chemistry simulation. It is therefore difficult to design an individual component to be as accurate as possible, when its accuracy is interlinked with the other components.

In [9], 10 properties are listed that are argued to be essential or highly desirable in making the dynamical core as accurate as possible, given restrictions on time and computational resources. Alongside this, the models should possess a certain amount of scalability on massively parallel computers: as the number of processes is increased the model run time should decrease. The properties proposed by [9] are:

1. mass conservation;
2. accurate representation of balanced flow and adjustment;
3. computational modes should be absent or well controlled;
4. the geopotential gradient and pressure gradient should produce no unphysical source of vorticity;
5. terms involving the pressure should be energy conserving;

6. Coriolis terms should be energy conserving;
7. there should be no spurious fast propagation of Rossby modes; geostrophic balance should not spontaneously break down;
8. axial angular momentum should be conserved;
9. numerical accuracy should approach second order;
10. minimal grid imprinting.

One of the most crucial of these properties is the second. Large scale atmospheric flow is close to geostrophic and hydrostatic balance. The numerical discretisation should therefore preserve a discrete version of these balances. The mechanism by which the atmosphere returns to geostrophic balance after a perturbation is known as *adjustment*, and takes place through the radiation and dispersion of the fast acoustic and inertial-gravity waves. The continuous shallow-water equations possess three sets of wave modes that should therefore be properly represented in order to accurately capture the balance flow and adjustment: these are a set each of eastward- and westward-propagating fast inertial-gravity waves, and the westward-propagating slow Rossby waves.

3.1.3 Review of Horizontal Grids

A latitude-longitude grid is used by many operational weather and climate models as its properties (logically rectangular structure, symmetry and orthogonality) can be manipulated to obtain the 10 properties above [9]. One of the vital aspects of a choice of grid is the choice of DOFs – the points at which the variables such as velocity and pressure are evaluated. There are two sides to this: the number of DOFs for each variable, and their staggering relative to one another. The choice in arrangement of DOFs turns out to be critical in governing the properties that the grid will have. When choosing a horizontal grid, one of the key aspects to consider is the arrangement of the DOFs. Five of the obvious choices of staggering for a rectangular grid were originally investigated by [47], [48] and [49]. These papers define the labelling for the staggerings that has since been used in the literature. This labelling can be seen in Figure 3.1.

The early work of [47], [48] and [49] analysed some of the properties of these discretisations applied to the shallow-water equations. The dispersion properties of these grids were evaluated by [50], which looked at the phase and group velocities of perturbations to the shallow-water equations in the f-plane (the regime in which the Coriolis parameter is kept constant). This work was extended by [51] to include an unstaggered vorticity-divergence formulation known as Z-grid. However, on the f-plane, Rossby wave modes are degenerate and stationary, and as a result it is not possible to identify spurious computational Rossby modes [9]. Investigations were done on the β-plane by [52], while [53]

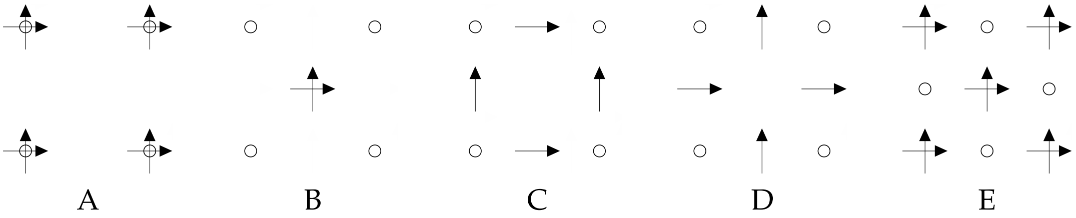


Figure 3.1: A key aspect of the horizontal discretisation used for the atmosphere is the placement of the degrees of freedom for the velocity and pressure fields. The most simple arrangements, which are shown here, are known as the Arakawa grids, with each designated by its own letter. Horizontal arrows represent the placement of the zonal wind degrees of freedom, whilst the vertical arrows represent the meridional wind degrees of freedom. The circles show the placement of the pressure degrees of freedom.

and [54] investigate dispersion on the spherical C-grid.

This research has all contributed to the argument that the C-grid and Z-grid are the best staggerings for representing the dispersion properties of inertia-gravity waves and Rossby waves, and hence also the best for representing balanced flow and adjustment on a quadrilateral grid. These grids also do not allow spurious pressure modes which are oscillatory and yet have a vanishing discrete gradient. As the Z-grid uses vorticity and velocity-divergence as its variables, each time step requires the solution of Poisson equations on each model level to find the wind field [9]. The feasibility of the Z-grid depends upon a fast and scalable Poisson solver. In the absence of this, a C-grid discretisation is therefore the best option for quadrilateral grids, and has been the best grid at providing the desired properties described above, and so is widely used operationally.

In [9], the authors also review the properties of non-quadrilateral grids. One key issue that is discussed by [9] and [8] is the effect of the ratio of mass (or density or pressure) degrees of freedom to those of velocity (or momentum). Grids made up of cells with a different number of edges will have different ratios of DOFs. It was observed by [55] that the amount of noise in a representation of geostrophic balance in different finite element discretisations depended upon this ratio. [8] explains that the DOF ratio defines the ratio of inertia-gravity modes to Rossby modes that the discretisation will support. Hence, if the ratio of velocity to pressure DOFs is not correct, the grid will support spurious computational modes. The C-grid on quadrilaterals has two velocity DOFs for every pressure DOF, which is considered the ideal ratio. In contrast, attempts to replicate C-grid staggering on triangular cells gives three velocity DOFs for every two pressure DOFs, whilst C-grid hexagonal cells have a ratio of 3:1 velocity-to-pressure DOFs.

The constraint provided by the requirement of a 2:1 velocity-to-pressure DOF ratio heavily restricts the choices of grid. Although quadrilaterals can be easily formed by subdividing other polygons, such as triangles, pentagons and hexagons, the resulting quadrilaterals will not preserve the orthogonality attribute of the latitude-longitude grid thought to be very important in deriving some of the desired properties listed in Section 3.1.2, such as the ability to design a mass-conserving transport scheme [9]. For these reasons, a latitude-longitude grid with C-grid staggering has been the preferred choice of grid for many dynamical cores.

3.1.4 The Pole Problem

As described by [9], the latitude-longitude grid also has drawbacks. The convergence of meridians at the poles means that cell sizes near the pole will be much smaller than those at the equator. This is particularly true as the model resolution is increased, as it must be in order to improve the model's accuracy. However, explicit time stepping schemes with Eulerian advection place restrictions upon the size of the time step: this is known as the Courant-Friedrichs-Lewy (CFL) condition. Hence, as the cell size shrinks at the poles, the time step used in the model must also shrink, which heavily restricts the resolution that can be used given a certain amount of computational resources.

Although this problem was bypassed by the advent of implicit time stepping and semi-Lagrangian advection, the 'pole problem' is beginning to re-emerge, as described by [9]. This time it is related to *parallel scalability* – the improvement in performance that is achieved with an increase in available computational resources. There are two conventional ways of measuring this parallel performance: *strong scalability*, which measures how the computation time varies with the number of processors for a computational problem of a fixed total size; and *weak scalability*, which measures how the computation time varies with the number of processors for a problem of a fixed size per processor.[*]

Whereas in the past, computing power has improved through advances in processor speed, it is expected to improve in the future by the development of even more massively-parallel machines, which use processors that are similar speed to current ones [56]. A key concept in adapting a model to parallel computing is to simultaneously solve the equations for different parts of the grid on different parts of the computer. However as the resolution of a latitude-longitude grid is increased, this will require increasing amounts of data communication between the processors dealing with the points around the poles. It had been previously anticipated that eventually the speed of this data communication will provide a bottleneck on the performance of the model, and improvements in parallel computing would not allow models of increased resolution to become feasible. Indeed, limitations in scalability have already been observed in the UK Met Office model [57]. More information about the development of algorithms for future computational architectures can be found in [58].

[*] Neither of these types of scalability exactly describes what is necessary for operational weather forecasters, who have a fixed wallclock time and want to know how the size of the computational problem can be varied with the number of available processors.

Thus, there is an increasing interest in different quasi-uniform grids that would not suffer from the 'pole problem'. Recent research into mixed finite element methods by [8] has shown that these can provide many of the desired properties listed in Section 3.1.2. These methods provide a way to control the number of DOFs per element, and hence to avoid spurious inertia-gravity or Rossby modes and to replicate many of the positive attributes of the C-grid staggering.

3.1.5 The GungHo Project

In [8], Cotter and Shipton show that compatible finite element methods can be used to discretise the linear rotating shallow-water equations while preserving some of the most important properties listed in Section 3.1.2. In particular, they showed that mass and energy can be conserved, and that following the work of [33], the commutativity of diagram (2.26) can be used to ensure the accurate representation of balanced geostrophic modes and the absence of spurious pressure modes. They showed that if the function space choices for the velocity $\mathbf{u}$ and the height[†] field η have a DOF-ratio of 2:1, the discretisation will not generate spurious branches of Rossby wave modes or inertia-gravity wave modes. [8] then gave the example of the spaces from the de Rham complex (2.22b) which satisfy these conditions.

Crucially, compatible finite element methods can provide these properties without relying on the orthogonality of the mesh, and can more easily facilitate other meshes that are quasi-uniform over the sphere. For instance, a cubed-sphere mesh can be used, which is illustrated in Figure 3.2. For these reasons, the next-generation dynamical core of the UK Met Office will use a compatible finite element discretisation. The larger project of developing such a dynamical core is called GungHo.

Other research into the use of compatible finite elements methods for numerical weather prediction includes [59, 60, 61, 62, 63, 64, 65, 10].

3.2 Gusto

Gusto is a dynamical core library that harnesses the Firedrake software to provide compatible finite element discretisations for atmospheric flow. It intends to provide an abstraction of the dynamical core, with the user able to choose which equation set to solve, as well as other aspects of the discretisation: time integration schemes, transport schemes, finite element spaces, among others. In this section we give an overview of these options for solving the compressible Euler equations (3.1). These options are used in the following two chapters, which revolve around the implementation of new fea-

[†] In the shallow-water equations, the height field η also corresponds to the pressure field.

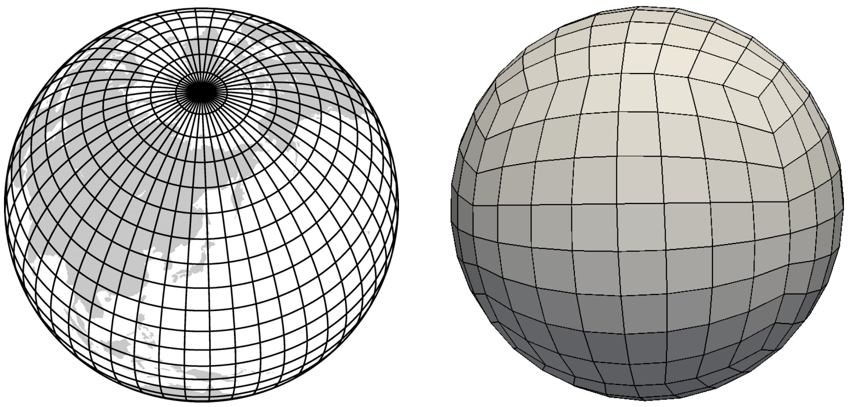


Figure 3.2: (Left) a latitude-longitude grid, whose meridians converge at the poles, causing very small cell sizes. This provides constraints on the rate of data communication in massively parallel supercomputers – the so-called 'pole problem'. (Right) a cubed-sphere mesh, which is an alternative quasi-uniform mesh over the sphere.

tures within Gusto. Chapter 4 presents a new transport scheme facilitating the use of the lowest-degree sets of finite element spaces, while Chapter 5 describes the development of moisture within Gusto.

3.2.1 Overview

Here we present an outline of our model structure, which is based on the UK Met Office ENDGame model [66]. We illustrate this with some pseudocode that describes how we evolve our prognostic variables, which we denote by the single vector $\chi = (\boldsymbol{v}, \rho_{\mathrm{d}}, \theta_{\mathrm{d}})$. Overall the model could be loosely described as a semi-implicit *predictor-corrector*. The operators evolving the prognostic variables are split into explicit steps, in which the model state at the next time step is 'predicted' by advection from a mean flow. This is then 'corrected' by an implicit step, which iteratively solves linearised equations. Within a time step these stages form an outer predictor loop and an inner corrector loop to increase stability whilst describing the fast gravity waves.

In the code below, we use $\mathcal{F}(\chi)$ to describe the forcing operation which is detailed in Section 3.2.4. The explicit application of the forcing operator is performed at the start of the time step. The 'predicting' advection step uses the advecting velocity $\overline{\mathbf{u}}$ to evolve the model, an operation we denote by a $\mathcal{A}_{\overline{\mathbf{u}}}$. The 'correcting' step first applies the forcing operator to the latest best-guess of the state, trying to solve the implicit part of the model. The difference between the implicit forcing term and the other terms gives the residual $\Delta\chi$. Reducing the residual to zero then ensures the implicit part of the model is solved. This is done by approximating the system as a linear problem $\mathcal{S}(\chi')$ for a perturbation χ'. Then the residual is reduced by iteratively solving for χ' and uses this to increment the best guess of χ at the next time level. At the end of the time step, we have the option to apply an artificial diffusion scheme, which we represent with $\mathcal{D}(\chi)$. In Chapter 5, we will add physical parametrisations in a similar way to this. The parameter α can be varied to off-center the time stepping, although we will take $\alpha = 1/2$.

The pseudocode summarising the algorithm is:

```
1. FORCING: $\chi^* = \chi^n + (1-\alpha)\Delta t \mathcal{F}(\chi^n)$
2. SET: $\chi_p^{n+1} = \chi^n$
3. OUTER:
    (a) UPDATE: $\bar{\mathbf{u}} = \alpha \boldsymbol{v}_p^{n+1} + (1-\alpha)\boldsymbol{v}^n$
    (b) ADVECT: $\chi_p = \mathcal{A}_{\bar{\mathbf{u}}}(\chi^*)$
    (c) INNER:
        i. FIND RESIDUAL: $\Delta\chi = \chi_p + \alpha\Delta t \mathcal{F}(\chi_p^{n+1}) - \chi_p^{n+1}$
        ii. SOLVE: $\mathcal{S}(\chi') = \Delta\chi$ for $\chi'$
        iii. INCREMENT: $\chi_p^{n+1} = \chi_p^{n+1} + \chi'$
4. DIFFUSION: $\chi_p^{n+1} = \mathcal{D}(\chi_p^{n+1})$
5. ADVANCE TIME STEP: $\chi^n = \chi_p^{n+1}$
```

3.2.2 Spaces

One of the main results of [8] was the proposal of combinations of function spaces for the velocity $\boldsymbol{v}$ and the (dry) air density ρ_d within this compatible finite element framework that will satisfy the 2:1 DOF ratio condition. In this section we will expand on this to describe the function spaces (V_v, V_ρ, V_θ) that we use in our model for velocity, density and potential temperature respectively, for both the two-dimensional vertical slice and the three-dimensional configurations that we use. Other papers describing these spaces include [62, 63, 65, 10].

A challenge associated with geophysical fluid dynamics is the high aspect ratio of the fluid, which motivates us to discretise our domain Ω using a mesh with regular vertical layers. Such a mesh is formed by *extruding* some base horizontal mesh over horizontal domain Ω_H with a one-dimensional vertical domain $\Omega_V \subset \mathbb{R}^1$. On such a mesh, finite elements can be formed from simpler finite elements by taking their tensor product, as described in [22]. In this language, the domain $\Omega = \Omega_H \otimes \Omega_V$. This is the approach that we will use, following the implementation described in [22]. Another advantage of using tensor product spaces is that it facilitates using spaces of different degree in the vertical direction to the horizontal, which may be of interest in the future.

One final ingredient that it is necessary to introduce is the HDiv operation, which forms a vector-valued $H(\mathrm{div})$ element from a tensor product element, as described in [22].

For now we wish to run our model both with two-dimensional vertical slices and in three-dimensions with a structured quadrilateral mesh. To form the tensor product spaces on these domains, we must first list the underlying compatible finite element spaces. We repeat here the one- and two-dimensional complexes from equation (2.22) in Section 2.3.2:

$$0 \to Q_{k+1}(\mathbb{R}^1) \xrightarrow{d/dx} dQ_k(\mathbb{R}^1) \to 0, \tag{3.7a}$$

$$0 \to Q_{k+1}(\mathbb{R}^2) \xrightarrow{\boldsymbol{\nabla}^{\perp}} RTc_k^f(\mathbb{R}^2) \xrightarrow{\boldsymbol{\nabla}\cdot} dQ_k(\mathbb{R}^2) \to 0. \tag{3.7b}$$

The compatible spaces used for two-dimensional vertical slices are formed by taking the tensor product of spaces in the one-dimensional complex (3.7a). For V_θ, we mimic the Charney-Philips grid by co-locating temperature DOFs with those of vertical velocity (and thus staggering them with the density DOFs). This is the same approach as described and motivated in [62] and [65].

To form spaces we use with horizontal degree k and vertical degree l are thus

$$V_v = \mathtt{HDiv}\{Q_{k+1}(\Omega_H) \otimes dQ_l(\Omega_V)\} \oplus \mathtt{HDiv}\{dQ_k(\Omega_H) \otimes Q_{l+1}(\Omega_V)\}, \tag{3.8a}$$

$$V_\rho = dQ_k(\Omega_H) \otimes dQ_l(\Omega_V), \tag{3.8b}$$

$$V_\theta = dQ_k(\Omega_H) \otimes Q_{l+1}(\Omega_V). \tag{3.8c}$$

In the case that $k = l$, these tensor product spaces become equivalent to just the two-dimensional complex (3.7b) with no tensor product, if we take temperature to be in the same space as the vertical component of velocity.

The spaces used for three-dimensional simulations are the tensor product of the two-dimensional complex (3.7b) used for the horizontal discretisation with the one-dimensional complex (3.7a). These are

$$V_v = \mathtt{HDiv}\left\{RTc_k^f(\Omega_H) \otimes dQ_l(\Omega_V)\right\} \oplus \mathtt{HDiv}\{dQ_k(\Omega_H) \otimes Q_{l+1}(\Omega_V)\}, \tag{3.9a}$$

$$V_\rho = dQ_k(\Omega_H) \otimes dQ_l(\Omega_V), \tag{3.9b}$$

$$V_\theta = dQ_k(\Omega_H) \otimes Q_{l+1}(\Omega_V). \tag{3.9c}$$

Again, if $k = l$ these spaces are equivalent to those in the three-dimensional complex in the finite element family $\mathcal{Q}_r^- \Lambda^k$ listed in equation (2.22c).

For the remainder of this thesis, we use equal horizontal and vertical degrees. We use two cases, $k = l = 0$ and $k = l = 1$, which are sometimes respectively referred to as the 'lowest-order' spaces and the 'next-to-lowest-order' spaces. For these two cases, we use slightly different model configurations, which will be listed in the coming sections. Chapter 4 focuses on the implementation of the transport scheme for the $k = 0$ case. The spaces used in these cases are illustrated in Table 3.1.

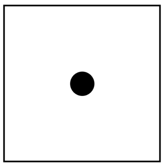
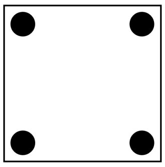
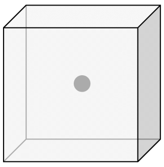
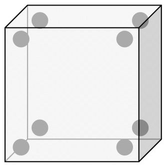
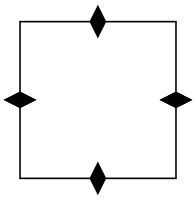
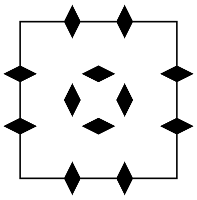
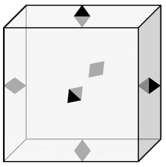
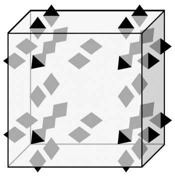
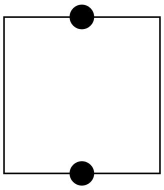
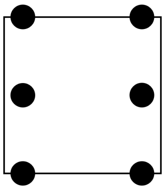
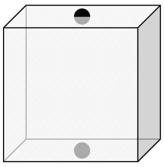
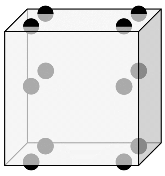
Space	$k = 0, d = 2$	$k = 1, d = 2$	$k = 0, d = 3$	$k = 1, d = 3$
V_ρ				
V_v				
V_θ				

Table 3.1: An illustration of the function spaces that we use for density, velocity and potential temperature variables in vertical slice ($d = 2$) and three-dimensional ($d = 3$) simulations. We have have two configurations for each case: with vertical and horizontal degree of $k = 0$ and $k = 1$. These spaces are tensor product elements listed in (3.8a) and (3.9a).

3.2.3 Differences Between $k = 0$ and $k = 1$ Configurations

Our model is designed for use with both the $k = 0$ and $k = 1$ sets of function spaces. The formulation is slightly different between these two cases, and here we give a summary of those differences, which is also displayed in Table 3.2.

The main area of difference is in the advection part of the model. In the $k = 0$ case, we use the recovered space transport schemes that were presented in [13] and are the focus of Chapter 4. However when using the $k = 1$ spaces, a discontinuous Galerkin upwind scheme is used for the transport of ρ_d, whilst the advection of θ_d is performed by the embedded scheme introduced by [67]. The velocity $\boldsymbol{v}$ advection equation is written in vector invariant form, and uses the theta time stepping method. All other advection schemes use a three-step Runge-Kutta time stepping procedure.

Component	$k = 0$ configuration	$k = 1$ configuration
Forcing	Non-advective dynamics terms	
Transport of $\boldsymbol{v}$	Recovered DG upwinding, SSPRK-3 time stepping	Vector invariant form, theta method time stepping
Transport of ρ_d	Recovered DG upwinding, SSPRK-3 time stepping	DG upwinding, SSPRK-3 time stepping
Transport of θ_d	Recovered DG upwinding, SSPRK-3 time stepping	Embedded DG upwinding, SSPRK-3 time stepping
Linear solve	Hybridised linear solver	
Diffusion	Recovered diffusion	Interior penalty

Table 3.2: A summary of the components of the different model configurations when using Gusto with the lowest order $k = 0$ spaces, and the next-to-lowest order $k = 1$ spaces. The advection schemes are a major source of difference between the two configurations.

3.2.4 Forcing

The 'forcing' operation involves finding the level of balance between the non-advective terms in (3.1a) when it is written in weak form. This imbalance is written as $\mathcal{F}(\boldsymbol{v})$.

For both $k = 0$ and $k = 1$ configurations, the forcing that we apply to $\boldsymbol{v}$ is the solution $\boldsymbol{v}_{\text{trial}} \in V_v$, for all $\boldsymbol{\psi} \in V_v$, to

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \boldsymbol{v}_{\text{trial}} \, \mathrm{d}x = & \int_\Omega \left[c_{pd} \boldsymbol{\nabla} \cdot (\theta_d \boldsymbol{\psi}) \Pi - \boldsymbol{\psi} \cdot (\boldsymbol{f} \times \boldsymbol{v}) - g(\boldsymbol{\psi} \cdot \widehat{\boldsymbol{k}}) \right] \mathrm{d}x \\ & - \int_\Gamma c_{pd} \, [\![\theta_d \boldsymbol{\psi}]\!]_{\boldsymbol{n}} \, \langle \Pi \rangle \, \mathrm{d}S,\end{aligned} \tag{3.10}$$

where Ω is the domain, Γ is the set of all interior facets and the angled brackets $\langle \cdot \rangle$ denote the average value on either side of a facet. The double square brackets with respect to $\widehat{\boldsymbol{n}}$, denoted by $[\![\cdot]\!]_{\boldsymbol{n}}$, represents the jump of vector $\boldsymbol{\psi}$ given by[‡]

$$[\![\boldsymbol{\psi}]\!]_{\boldsymbol{n}} = \boldsymbol{\psi}^+ \cdot \widehat{\boldsymbol{n}}^+ + \boldsymbol{\psi}^- \cdot \widehat{\boldsymbol{n}}^-. \tag{3.11}$$

The forcing term $\mathcal{F}(\boldsymbol{v})$ is then taken to be the solution $\boldsymbol{v}_{\text{trial}}$ to this.

3.2.5 Advection

In the advection stage, each of the variables is translated by the velocity $\overline{\boldsymbol{u}}$. We represent the action of this by the operator $\mathcal{A}_{\overline{\boldsymbol{u}}}$.

[‡] We attempt to distinguish this from the jump with subscript $+$, which for vector $\boldsymbol{\psi}$ is given by $[\![\boldsymbol{\psi}]\!]_+ = \boldsymbol{\psi}^+ - \boldsymbol{\psi}^-$.

Embedded DG Advection

Here we briefly outline the embedded DG transport scheme introduced in [67]. First, we remind ourselves of the single forward-Euler step of discontinuous Galerkin upwinding. This was presented in equation (2.12) of Section (2.2.2), and here we describe it as the operation $\mathcal{L}_{\bar{\mathbf{u}}}$ upon a quantity $\mathbf{q}$, which may be scalar or vector. For the advective form of the transport equation this involves finding the solution $\mathbf{q}_{\text{trial}}$, for all $\boldsymbol{\psi} \in V_q$ of

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \mathbf{q}_{\text{trial}} \, dx - \int_\Omega \boldsymbol{\psi} \cdot \mathbf{q} \, dx - \Delta t \int_\Omega \mathbf{q} \cdot [\boldsymbol{\nabla} \cdot (\boldsymbol{\psi} \otimes \bar{\mathbf{u}})] \, dx \\ + \Delta t \int_\Gamma \left(\bar{\mathbf{u}} \cdot \hat{\mathbf{n}}^+\right) \mathbf{q}^\dagger \cdot [\![\boldsymbol{\psi}]\!]_+ \, dS = 0.\end{aligned} \tag{3.12}$$

The difference between equation (3.12) and (2.12) is the possibility that $\boldsymbol{\psi}$ and $\mathbf{q}$ can be vector quantities, which requires the introduction of the outer product $\otimes$. Otherwise, $\mathbf{q}^\dagger$ denotes the upwind value of $\mathbf{q}$ from equation (2.11). As before in (2.10), the jump $[\![\cdot]\!]_+$ is taken with respect to the positively labelled side of the facet.

We also need to solve the advection equation in continuity form, as in the transport of ρ_d. The single-step forward-Euler equation for the continuity form of the transport equation is

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \mathbf{q}_{\text{trial}} \, dx - \int_\Omega \boldsymbol{\psi} \cdot \mathbf{q} \, dx - \Delta t \int_\Omega (\bar{\mathbf{u}} \otimes \mathbf{q}) : (\boldsymbol{\nabla} \otimes \boldsymbol{\psi}) \, dx \\ + \Delta t \int_\Gamma \left(\bar{\mathbf{u}} \cdot \hat{\mathbf{n}}^+\right) \mathbf{q}^\dagger \cdot [\![\boldsymbol{\psi}]\!]_+ \, dS = 0,\end{aligned} \tag{3.13}$$

where $\mathbf{A} : \mathbf{B}$ represents the Frobenius inner product between $\mathbf{A}$ and $\mathbf{B}$.

Both equations (3.12) and (3.13) describe a single step which is unstable for discontinuous linear polynomials as the cell size $\Delta x \to 0$. Stability and increased accuracy with respect to Δt is achieved by composing these steps using the three-step Runge-Kutta scheme from Section 2.2.2 (which we call SSPRK-3).

This DG upwinding scheme can be used directly for discontinuous spaces, such as that used for ρ_d. The embedded DG scheme presented in [67] was motivated for advecting functions in V_θ where $k = 1$. This space is quadratic in the vertical and continuous between elements, whilst it is discontinuous linear in the horizontal. This space is embedded within the space with the same horizontal structure that is quadratic in the vertical but discontinuous between elements. The advection scheme of [67] performs a single time step of advection by injecting the field into this embedding space, where transport by $\mathcal{A}_{\bar{\mathbf{u}}}$ takes place. The field is then returned to its original space via a Galerkin projection. A representation of this scheme is given in Figure 3.3.

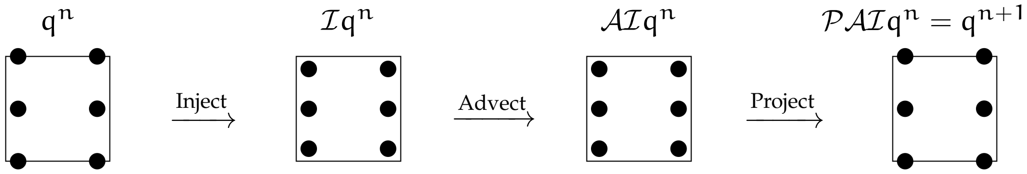

q^n
Inject
$\mathcal{I}q^n$
Advect
$\mathcal{A}\mathcal{I}q^n$
Project
$\mathcal{P}\mathcal{A}\mathcal{I}q^n = q^{n+1}$

Figure 3.3: A diagram illustrating the steps in the embedded DG upwinding scheme presented in [67], which is designed for advecting the potential temperature θ_d. In the $k = 1$ configuration, this is in a space which is discontinuous and linear in the horizontal but quadratic and continuous in the vertical. The steps of the scheme, transforming a variable $q^{(n)}$ at the n-th time step, are to a) inject the variable into the broken space; b) to advect the field in the broken space, using a DG upwinding scheme; c) to project the field back into the original space.

Advection with the $k = 1$ spaces

For the $k = 1$ case, ρ_d is advected by using the simple DG upwinding scheme (3.13) for the continuity form of the advection equation, within the SSPRK-3 time stepping scheme of equation (2.13) presented in [28].

The θ_d field is transported using the embedded DG scheme of [67] that was outlined in Section 3.2.5, using the DG upwinding scheme (3.12) again with the SSPRK-3 time stepping scheme. The embedding space is quadratic in the vertical and linear in the horizontal.

The most notable difference in the advection stage of the model between using the $k = 0$ and $k = 1$ spaces is the transport of the velocity. For the $k = 1$ configuration, we write the advection equation in vector-invariant form, which in three dimensions involves discretising

$$\frac{\partial \boldsymbol{v}}{\partial t} + (\boldsymbol{\nabla} \times \boldsymbol{v}) \times \overline{\boldsymbol{u}} + \frac{1}{2}\boldsymbol{\nabla}(\boldsymbol{v} \cdot \overline{\boldsymbol{u}}) = 0. \tag{3.14}$$

The right hand side of this equation appears in the forcing part stage of the model. The action of advection operator $\mathcal{L}_{\bar{u}}$ gives the solution $\boldsymbol{v}_{\text{trial}} \in V_v$ for all $\boldsymbol{\psi} \in V_v$ to

$$\begin{aligned} &\int_\Omega \boldsymbol{\psi} \cdot (\boldsymbol{v}_{\text{trial}} - \boldsymbol{v}) \, dx - \int_\Gamma \boldsymbol{v}^\dagger \cdot \left[\widehat{\boldsymbol{n}}^+ \times [\![\overline{\boldsymbol{u}} \times \boldsymbol{\psi}]\!]_+\right] dS \\ &\quad + \int_\Omega \boldsymbol{v} \cdot [\boldsymbol{\nabla} \times (\overline{\boldsymbol{u}} \times \boldsymbol{\psi})] \, dx - \frac{1}{2}\int_\Omega (\boldsymbol{v} \cdot \overline{\boldsymbol{u}})(\boldsymbol{\nabla} \cdot \boldsymbol{\psi}) \, dx = 0, \end{aligned} \tag{3.15}$$

with $\boldsymbol{v}^\dagger$ as the upwind value of $\boldsymbol{v}$ and $[\![\cdot]\!]_+$ taking the same definition as in Section 3.2.5. In two dimensions, this becomes

$$\begin{aligned} &\int_\Omega \boldsymbol{\psi} \cdot (\boldsymbol{v}_{\text{trial}} - \boldsymbol{v}) \, dx - \int_\Gamma \left(\boldsymbol{v}_\perp^\dagger \cdot \widehat{\boldsymbol{n}}^+\right) [\![\overline{\boldsymbol{u}}_\perp \cdot \boldsymbol{\psi}]\!]_+ \, dS \\ &\quad - \int_\Omega \boldsymbol{v} \cdot \boldsymbol{\nabla}^\perp (\boldsymbol{\psi} \cdot \overline{\boldsymbol{u}}_\perp) \, dx - \frac{1}{2}\int_\Omega (\boldsymbol{v} \cdot \overline{\boldsymbol{u}})(\boldsymbol{\nabla} \cdot \boldsymbol{\psi}) \, dx = 0, \end{aligned} \tag{3.16}$$

where the perpendicular of vector $\mathbf{f} = (f_x, f_y)$ is given by

$$\mathbf{f}_\perp = (-f_y, f_x). \tag{3.17}$$

The theta time stepping scheme is used, so that the velocity at the $(n+1)$-th time step is found by solving

$$\boldsymbol{\nu}^{(n+1)} = \boldsymbol{\nu}^{(n)} + \Delta t \left[\vartheta \mathcal{L}_{\bar{u}} \boldsymbol{\nu}^{(n)} + (1-\vartheta)\mathcal{L}_{\bar{u}} \boldsymbol{\nu}^{(n+1)}\right] \tag{3.18}$$

We take $\vartheta = 1/2$, making this a Crank-Nicolson scheme [68].

ADVECTION WITH THE $k = 0$ SPACES

The DG upwinding schemes used for the $k = 1$ configuration do not have second-order accuracy for the lowest-order spaces. Instead, we use the recovered transport schemes introduced in [13] and detailed in Chapter 4.

3.2.6 LINEAR SOLVE

We now present the strategy for the implicit stage of the model. When initialising the model, we set mean ρ_d and θ_d states which we denote respectively as $\bar{\rho}_d$ and $\bar{\theta}_d$. The residual $\Delta\chi$ between the predicted explicit χ_p and the latest guess χ_p^{n+1} of the prognostic variables is computed first. Reducing this residual to zero solves the implicit part of the model. This is done by solving a linear set of equations, with the residual on the right hand side, to calculate increments that are added to χ_p^{n+1}, updating it.

The linearised problem that we are attempting to solve is

$$\boldsymbol{\nu}' + \alpha\Delta t c_{pd}\left(\theta_d' \boldsymbol{\nabla}\bar{\Pi} + \bar{\theta}_d \boldsymbol{\nabla}\Pi'\right) = \Delta\boldsymbol{\nu} \tag{3.19a}$$

$$\rho_d' + \alpha\Delta t \boldsymbol{\nabla}\cdot\left(\bar{\rho}_d \boldsymbol{\nu}'\right) = \Delta\rho_d \tag{3.19b}$$

$$\theta_d' + \alpha\Delta t\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\bar{\theta}_d\right)\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{\nu}'\right) = \Delta\theta_d, \tag{3.19c}$$

where the primes represent perturbations. This comes from linearising the dry equations.

These equations will be solved in weak form. However, we simplify them first by eliminating θ_d' using the strong form equation. This allows us to create a hybridised mixed system to be solved for $\boldsymbol{\nu}'_{\text{trial}}$ and ρ'_{trial}. Taking $\widehat{\boldsymbol{k}}$ as the unit upward normal and $\Delta\theta_d$ as the residual of θ_d, we introduce

$$\Theta' = \Delta\theta_d - \alpha\Delta t\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\bar{\theta}_d\right)\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{\nu}'_{\text{trial}}\right), \tag{3.20}$$

where α is the time stepping parameter. The Exner pressure perturbation is then approximated as

$$\Pi' = \frac{\kappa}{1-\kappa}\overline{\Pi}\left(\frac{\Theta'}{\overline{\theta}_d} + \frac{\rho'_{\text{trial}}}{\overline{\rho}_d}\right), \tag{3.21}$$

where

$$\overline{\Pi} = \left(\frac{\overline{\rho}_d\overline{\theta}_d R_d}{p_R}\right)^{\frac{\kappa}{1-\kappa}}. \tag{3.22}$$

The hybridised mixed system involves breaking the continuity of normal components of $\boldsymbol{v}$ between elements, introducing the broken space $\widehat{V}_{\boldsymbol{v}}$. In conjunction, a trace field ℓ_{trial} is introduced, approximating the average values of Π perturbations on the trace space V_{trace}, which is the discontinuous space of functions that live on the boundary of $V_{\boldsymbol{v}}$. This variable acts as Lagrange multipliers to provide the continuity constraint of the pre-hybridised system. The resulting mixed system for $(\boldsymbol{v}'_{\text{trial}}, \rho'_{\text{trial}}, \ell'_{\text{trial}}) \in \left(\widehat{V}_{\boldsymbol{v}}, V_\rho, V_{\text{trace}}\right)$, which holds for all $(\boldsymbol{\psi}, \phi, \lambda) \in \left(\widehat{V}_{\boldsymbol{v}}, V_\rho, V_{\text{trace}}\right)$, can be written as

$$\begin{aligned}
&\int_\Omega \boldsymbol{\psi}\cdot\left(\boldsymbol{v}'_{\text{trial}} - \Delta\boldsymbol{v}\right)\,dx \\
&- \alpha\Delta t c_{pd}\left(\int_\Omega \boldsymbol{\nabla}\cdot\left[\Theta' w\widehat{\boldsymbol{k}}\right]\overline{\Pi}\,dx - \int_\Gamma \left[\!\left[\Theta'\widehat{\boldsymbol{k}}\right]\!\right]_{\boldsymbol{n}}\left\langle\overline{\Pi}\right\rangle\,dS\right) \\
&+ \alpha\Delta t c_{pd}\left(\int_{\partial\Omega} \Theta' w\widehat{\boldsymbol{k}}\cdot\widehat{\boldsymbol{n}}\left\langle\overline{\Pi}\right\rangle\,ds - \int_\Omega \boldsymbol{\nabla}\cdot\left[\overline{\theta}_d\boldsymbol{\psi}\right]\Pi'\,dx\right) \\
&+ \alpha\Delta t c_{pd}\left(\int_\Gamma \left[\!\left[\overline{\theta}_d\boldsymbol{\psi}\right]\!\right]_{\boldsymbol{n}}\ell'_{\text{trial}}\,dS + \int_{\partial\Omega}\overline{\theta}_d\left(\boldsymbol{\psi}\cdot\widehat{\boldsymbol{n}}\right)\ell'_{\text{trial}}\,ds\right) \\
&+ \int_\Omega \phi\left(\rho'_{\text{trial}} - \Delta\rho_d\right)\,dx - \alpha\Delta t\int_\Omega\left(\boldsymbol{\nabla}\phi\cdot\boldsymbol{v}'_{\text{trial}}\right)\overline{\rho}_d\,dx \\
&+ \alpha\Delta t\left(\int_\Gamma \left[\!\left[\phi\boldsymbol{v}'_{\text{trial}}\right]\!\right]_{\boldsymbol{n}}\left\langle\rho_d\right\rangle\,dS + \int_{\partial\Omega}\phi\boldsymbol{v}'_{\text{trial}}\cdot\widehat{\boldsymbol{n}}\left\langle\rho_d\right\rangle\,ds\right) \\
&+ \int_\Gamma \lambda\left[\!\left[\boldsymbol{v}'_{\text{trial}}\right]\!\right]_{\boldsymbol{n}}\,dS + \int_{\partial\Omega}\lambda\left(\boldsymbol{v}'_{\text{trial}}\cdot\widehat{\boldsymbol{n}}\right)\,ds = 0,
\end{aligned} \tag{3.23}$$

where w is the vertical component of $\boldsymbol{\psi}$ and $\partial\Omega$ is the external boundary of the domain. This mixed system can be statically condensed into a single system for the Lagrange multipliers, giving a single system to be solved. Once this system has been solved, the calculated value of ℓ' is used to find ρ'_d and then the broken $\boldsymbol{v}'$. The recovery operator described in Chapter 4 is then used to restore the continuous velocity $\boldsymbol{v}'$. Finally, the value of θ'_d is found as the θ'_{trial} that solves for all $\gamma \in V_\theta$

$$\int_\Omega \gamma\left[\theta'_{\text{trial}} - \Delta\theta_d + \alpha\Delta t\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{v}'\right)\left(\widehat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\overline{\theta}_d\right)\right] = 0. \tag{3.24}$$

The approach of assuming only vertical variation of θ_d in (3.19c) introduces a small error. Another error will occur near topography, as the use of the strong equation to eliminate θ_d neglects the Jacobian terms necessary for the Piola transform used for computations over non-regular elements.

3.2.7 Diffusion

The artificial diffusion terms are discretised with an interior penalty method, presented in [69]. They are treated in the model like physics parametrisations, and the only test case that we will use them in is described in Section 4.6.3.

Let the action of the diffusion step upon a (scalar or vector) quantity $\mathbf{q}$ be given by $\mathcal{D}(\mathbf{q})$. Then $\mathcal{D}(\mathbf{q})$ is found to be the solution $\mathbf{p} \in V_q$, for all $\boldsymbol{\psi} \in V_q$, to

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \mathbf{p} \, dx &= \int_\Omega \boldsymbol{\psi} \cdot \mathbf{q} \, dx + \Delta t \int_\Gamma 2 \left\langle \mathbf{p} \otimes \widehat{\mathbf{n}} \right\rangle : \left\langle K \boldsymbol{\nabla} \otimes \boldsymbol{\psi} \right\rangle \, dS \\ &+ \Delta t \int_\Gamma 2 \left\langle K \boldsymbol{\nabla} \otimes \mathbf{p} \right\rangle : \left\langle \boldsymbol{\psi} \otimes \widehat{\mathbf{n}} \right\rangle \, dS - \mu \Delta t \int_\Gamma 4 \left\langle \mathbf{p} \otimes \widehat{\mathbf{n}} \right\rangle : \left\langle K \boldsymbol{\psi} \otimes \widehat{\mathbf{n}} \right\rangle \, dS \\ &- \Delta t \int_\Omega 2 \left(\boldsymbol{\nabla} \otimes \boldsymbol{\psi} \right) : \left(K \boldsymbol{\nabla} \otimes \mathbf{p} \right) \, dx, \end{aligned} \tag{3.25}$$

where $\otimes$ represents the outer product of two vectors, $\langle \cdot \rangle$ is the average value on either side of a facet and $:$ is the Frobenius inner product. The penalty parameter μ is typically taken to be proportional to $1/\Delta x$. The inherent boundary conditions in this formulation are $\widehat{\mathbf{n}} \cdot \boldsymbol{\nabla} \otimes \mathbf{q} = \mathbf{0}$ on the boundary $\partial\Omega$. Other boundary conditions can be explicitly applied, such as $\mathbf{q} \cdot \widehat{\mathbf{n}} = 0$ on the boundary $\partial\Omega$, which ensures no outflow when the diffused quantity is the wind velocity $\boldsymbol{v}$. This is done by restricting $\mathbf{p}$ and $\boldsymbol{\psi}$ to lie in the subset of V_q with $\mathbf{q} \cdot \widehat{\mathbf{n}} = 0$.

The interior penalty approach has second-order numerical accuracy for the $k = 1$ spaces. However for the lowest-order $k = 0$ spaces, this is not the case. As with the transport schemes, we use a recovery scheme to augment equation (3.25) to obtain second order accuracy. This is presented in Chapter 4.

3.2.8 Hydrostatic Balance

An important step in setting up many test cases is to initialise hydrostatic balance, which must hold at the discrete level. This typically involves finding the ρ_d, given a θ_d profile, which gives rise to no vertical motions. The approach for obtaining a discrete hydrostatic balance in our model was presented in Section 7.3 of [62].

Introducing $V_{\boldsymbol{v}}^{\text{vert}}$, the density that gives hydrostatic balance is found from the $(\boldsymbol{w}, \rho_d) \in$

$\left(V_{\boldsymbol{v}}^{\text{vert}}, V_\rho\right)$ that solves, for all $(\boldsymbol{\psi}, \lambda) \in \left(V_{\boldsymbol{v}}^{\text{vert}}, V_\rho\right)$,

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \boldsymbol{w} \, \mathrm{d}x - \int_\Omega c_{pd} \Pi(\rho_d, \theta_d) \boldsymbol{\nabla} \cdot (\theta_d \boldsymbol{\psi}) = \\ - \int_{\partial\Omega_i} c_{pd} \Pi_R \theta_d \boldsymbol{\psi} \cdot \widehat{\boldsymbol{n}} \, \mathrm{d}s - \int_\Omega g \boldsymbol{\psi} \cdot \widehat{\boldsymbol{k}} \, \mathrm{d}x,\end{aligned} \tag{3.26a}$$

$$\int_\Omega \lambda \boldsymbol{\nabla} \cdot \boldsymbol{w} \, \mathrm{d}x = 0, \tag{3.26b}$$

subject to the boundary condition that $\boldsymbol{w} \cdot \widehat{\boldsymbol{n}} = 0$ on the boundary $\partial\Omega$, which ensures that $\boldsymbol{w}$ is zero. Π_R is a prescribed value of Π on the top or bottom boundary, $\partial\Omega_i$. The equation for $\boldsymbol{w}$ is introduced to produce a square system of equations

As a mixed discontinuous and continuous method, this can also be hybridised in the same way as the linear solver in Section 3.2.6. More information on the hybridization of mixed methods and its implementation can be found in [70, 40].

4 The Recovered Space Advection Scheme

Having motivated a dynamical core using a compatible finite element discretisation in Chapter 3, in Section 3.2 we presented the model details of Gusto, a dynamical core library currently being actively developed. In Section 3.2.2 we presented the compatible finite element spaces that are used in Gusto, focusing on two sets of spaces: the lowest-order $k = 0$ spaces and the 'next-to-lowest-order' $k = 1$ spaces. This chapter concerns the development of the advection schemes for the $k = 0$ spaces, which centres on the concept of recovering a higher-order representation of the field, before advecting in the higher-order space. The main implication of the results of this chapter is that different stages of discretisations can be performed with elements of different order, by using the recovery approach. The following results are presented in this chapter:

- the presentation of a new transport scheme, which facilitates the use of the lowest-order elements within Gusto;
- a proof of its stability, demonstrating that the concept of the scheme is valid;
- a description of the specific recovery operator that we use;
- a discussion of the stability properties of the transport scheme for three cases in one dimension, which represent three cases that we specifically use;
- numerical results showing the convergence properties of the transport scheme, demonstrating that the scheme has second-order numerical accuracy;
- demonstration of the advection scheme within Gusto, which represents the first results of Gusto solving the dry compressible Euler equations using the lowest-order elements;
- application of the recovery concept to a diffusion scheme, showing that this approach can be extended to more problems than advection alone;

This chapter is largely adapted from our paper [13], where we first presented these results.

4.1 Motivation

As was explained in Chapter 3, there has been recent interest in developing a dynamical core for numerical weather prediction using a compatible finite element discretisation.

This was shown by [8] to satisfy many of the important properties presented in [9] that we listed in Section 3.1.2. Crucially, these properties can be provided without requiring a latitude-longitude grid, so a quasi-uniform grid over the sphere can be facilitated and the scalability bottleneck associated with the poles on the latitude-longitude grid can be avoided.

In Section 3.2 we discussed Gusto – a library currently being developed of compatible finite element methods built upon the Firedrake software that can solve atmospheric equations. In particular, Section 3.2.2 presented two configurations that depended upon the degree of function space used. With the lowest-order $k = 0$ spaces, density is constant in each element (and discontinuous between them), while for the $k = 1$ spaces it is discontinuous and linear between elements. The spaces for all the variables are illustrated in Table 3.1.

One of the key motivations for using the $k = 0$ spaces is physics-dynamics coupling. The effects of the unresolved processes upon the resolved flow are usually calculated pointwise at the degrees of freedom. For fields that are piecewise constant or linear, the pointwise values of the field can be interpreted as the mean of the field within that element. As was demonstrated in Chapter 1, unresolved processes can be described by decomposing fields into fluctuating and mean parts, and thus there is a correspondence with this decomposition for constant or linear spatial discretisations of fields. For the $k = 1$ elements, the temperature field is piecewise quadratic in the vertical dimension and so the physical interpretation of the values of the degrees of freedom becomes less clear. Another advantage of the $k = 0$ spaces comes with the decreased overlap between basis functions when solving the linearised problem of the model (as described in Section 3.2.6), which can make the resulting linear system more sparse and hence easier to solve. Using the $k = 0$ spaces also provides a better direct comparison with other models, such as the one presented in [10].

The transport schemes that were described in Section 3.2.5 have second-order numerical accuracy when applied to the $k = 1$ spaces. In other words, the error between the discretised solution and a smooth true solution falls with $(\Delta x)^2$ as $\Delta x \to 0$ for cell length Δx. Numerical accuracy of second-order is one of the properties listed in 3.1.2 that were argued for by [9]. However, the schemes presented in 3.2.5 are not second-order accurate for the $k = 0$ spaces. Finite volume transport schemes can be used to attain second-order accuracy, with larger stencils used to better approximate numerical fluxes. Although this is similar to the approach used by [10], which uses a *method of lines* advection scheme, such procedures are currently not supported by Firedrake. Therefore a new transport scheme is necessary for the $k = 0$ spaces to be used within Gusto. Inspired by the recovery operator of [71], this chapter presents a new advection scheme for these lowest-order spaces.

After describing the scheme in Section 4.2, we discuss several of its properties in Section 4.3, including a general argument of its stability and von Neumann analysis of the scheme in three particular cases. Section 4.4 presents the results of numerical tests demonstrating the second-order numerical accuracy of the scheme, the stability calculations of Section 4.3.2 and the use of a limiter within the advection scheme. Finally, the use of the advection scheme within a model of the compressible Euler equations is presented in Section 4.6.

4.2 The 'Recovered Space' Scheme

The key ideas upon which this scheme is based are the embedded DG upwinding scheme of [67] and the family of recovered finite element methods introduced by [71]. These recovered methods combine features of discontinuous Galerkin approaches with conforming finite element methods. They are similar to other recovery methods, such as those in [72, 73], in that they reconstruct higher-order polynomials from lower order data in a patch of cells. They differ in that they do not attempt to reproduce polynomials of a certain degree exactly. Instead, they involve mapping discontinuous finite element spaces to continuous ones, via recovery operators, relying on analysis estimates of stability and accuracy. The scheme that we will introduce involves the use of one of these operators to recover a function in a discontinuous first-order space from one in a discontinuous zeroth-order space. To do this, we first recover a first-order continuous function from the zeroth-order discontinuous function using an averaging operator described in [71] and [74]. This operator finds the values for any degree of freedom shared between elements in a continuous function space, by averaging between the values of the surrounding degrees of freedom from the discontinuous space.

Once this operator has been applied, existing transport schemes can be used to perform the advection upon the recovered field. This approach is compatible when the transport equation is in *advective* form

$$\frac{\partial q}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla} q = 0, \tag{4.1}$$

or *conservative* form

$$\frac{\partial q}{\partial t} + \boldsymbol{\nabla} \cdot (q\boldsymbol{v}) = 0, \tag{4.2}$$

where q is the quantity to be transported by velocity $\boldsymbol{v}$. However most of our analysis will focus on the application of this scheme to the advective form of the equation, under which the mass $\int_\Omega q \, dx$ over the whole domain Ω will only be necessarily conserved when the flow is incompressible, $\boldsymbol{\nabla} \cdot \boldsymbol{v} = 0$.

4.2.1 The Scheme

First we will define a set of spaces that our functions will lie in. Let $V_0(\Omega)$ be the lowest-order finite element space in which the initial field lies, where Ω is our spatial domain*. $V_1(\Omega)$ is then the space of next degree, which will be fully discontinuous. We also have that $V_0 \subset V_1$. $\widetilde{V}_1(\Omega)$ is the fully continuous space of same degree as $V_1(\Omega)$, whilst $\widehat{V}_0(\Omega)$ is a broken (i.e. fully discontinuous) version of $V_0(\Omega)$. In many cases, $\widehat{V}_0(\Omega)$ and $V_0(\Omega)$ will coincide.

We now define a series of operators to map between these spaces.

Definition 1. *The recovery operator $\mathcal{R}$ acts upon a function in the initial space to make a function in the continuous space of higher degree, so that $\mathcal{R} : V_0 \to \widetilde{V}_1$. The operator has second-order numerical accuracy.*

Assumption 1. *The recovery operator $\mathcal{R}$ has the property that for all $q \in V_0$, there is some $C > 0$ such that $||\mathcal{R}q||_{L^2} \leqslant C||q||_{L^2}$.*

Definition 2. *The injection operator $\mathcal{I} : V \to V_1$ identifies a function in V_0, $\widehat{V}_0$ or $\widetilde{V}_1$ as a member of V_1. This must be numerically implemented, although it does nothing else mathematically.*

Definition 3. *The projection operator $\widehat{\mathcal{P}} : \widetilde{V}_1 \to \widehat{V}_0$ is defined to give $\widehat{u} = \widehat{\mathcal{P}}\widetilde{v}$, from $\widetilde{v} \in \widetilde{V}_1$, by finding the solution $\widehat{u} \in \widehat{V}_0$ to*

$$\int_\Omega \widehat{\psi}\widehat{u}\,dx = \int_\Omega \widehat{\psi}\widetilde{v}\,dx, \quad \forall\widehat{\psi} \in \widehat{V}_0. \tag{4.3}$$

Definition 4. *The advection operator $\mathcal{A} : V_1 \to V_1$ represents the action of performing one time step of a stable discretisation of the advection equation (in either advective or conservative form) and has second-order numerical accuracy in space.*

Definition 5. *The projection operator $\mathcal{P} : V_1 \to V_0$ will have two forms. The first, $\mathcal{P}_A$, is defined to give $u = \mathcal{P}_A v$ from $v \in V_1$, by finding the solution $u \in V_0$ to*

$$\int_\Omega \psi u\,dx = \int_\Omega \psi v\,dx, \quad \forall\psi \in V_0, \tag{4.4}$$

where $u \in V_0$ and $v \in V_1$.

The second form, $\mathcal{P}_B$, is composed of two operations: $\mathcal{P}_I : V_1 \to \widehat{V}_0$, interpolation into the broken space by pointwise evaluation at degrees of freedom, and $\mathcal{P}_R : \widehat{V}_0 \to V_0$, recovery from

*This spatial domain can be arbitrary, but with geophysical applications in mind we anticipate the scheme being used upon rectangular or cuboid domains (with or without periodicity) or spherical shells. However the recovery operator that we consider in Section 4.2.3 is intended for use in flat spaces or with only scalar fields in curved spaces and we do not yet consider the application to transport of vector fields in curved spaces. Therefore in this work we will predominantly consider rectangular domains with a vertical coordinate, with rigid walls at the top and bottom edges.

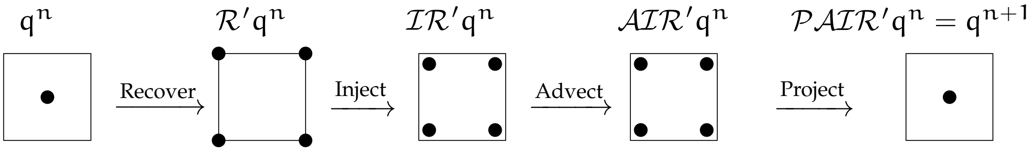


Figure 4.1: A diagram illustrating the steps in the 'recovered space' transport scheme, using the spaces presented in Table 4.1 for density ρ_d. The operator $\mathcal{R}'$ represents the corrected recovery operation $\mathcal{R}-\widehat{\mathcal{P}}\mathcal{R}+1$, where $\mathcal{R}$ is the recovery operator itself and the correction is done to ensure mass is conserved locally. After recovering $q^{(n)}$ and injecting it to V_1, it is advected by $\mathcal{A}$ within V_1. The value of q at the $(n+1)$-th time level is then the projection back into V_0.

the broken space to the original space, restoring continuity via the reconstruction operator from [71]. $\mathcal{P}_B$ *can thus be written as* $\mathcal{P}_B = \mathcal{P}_R\mathcal{P}_I$.

In the case that V_0 is fully discontinuous, $\mathcal{P}_A$ and $\mathcal{P}_B$ will be identical operations. However, for fully or partially continuous V_0, $\mathcal{P}_B$ prevents the formation of any new maxima and minima, whereas $\mathcal{P}_A$ does not. We may thus use $\mathcal{P}_B$ as the projection operator when trying to bound the transport, such as for a moisture species. Further discussion can be found in Section 4.2.4. The drawback is that whilst $\mathcal{P}_A$ preserves the mass (setting $\psi = 1$ gives $\int_\Omega u \, dx = \int_\Omega v \, dx$), $\mathcal{P}_B$ does not necessarily do so.

Definition 6. *The 'recovered space' scheme then takes the function* $q^n \in V_0$ *at the* n*-th time step and returns the function* $q^{n+1} \in V_0$ *at the* $(n+1)$*-th time step by performing the following series of operations:*

$$q^{n+1} = \mathcal{PAI}(\mathcal{R} - \widehat{\mathcal{P}}\mathcal{R} + 1)q^n, \tag{4.5}$$

where $\mathcal{P}$ *could be either* $\mathcal{P}_A$ *or* $\mathcal{P}_B$.

An important property of this scheme is that in the absence of flow, the field being advected must remain unchanged. In this case $\mathcal{A}$ will be the identity, and since $\mathcal{PIR} \equiv \mathcal{PI}\widehat{\mathcal{P}}\mathcal{R}$, then $q^{n+1} = \mathcal{PI}q^n = q^n$. In practice, mass will be only be conserved up to the precision used by the numerical solver for $\mathcal{P}$.

A diagram representing the scheme is shown in Figure 4.1.

4.2.2 Example Spaces

In this section we give an example set of spaces $\{V_0, V_1, \widetilde{V}_1, \widehat{V}_0\}$ on quadrilateral elements that can be used for this scheme, in the context of 2D vertical slice problems for the compressible Euler equations given in Section 3.1.1.

Since we are motivated by using the lowest-order family of compatible finite element spaces on extruded meshes of quadrilateral elements, that is what we describe here, along with the corresponding vertical slice spaces. For domains Ω_H and Ω_V, when taking the tensor product of two finite element spaces $U(\Omega_H)$ and $V(\Omega_V)$, the resulting

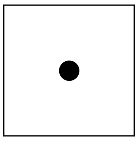
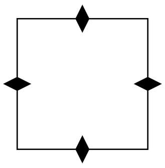
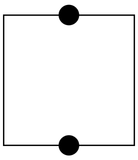
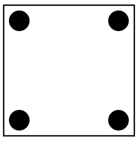
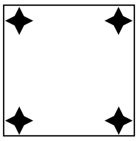
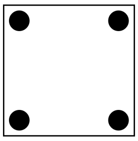
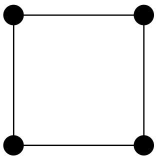
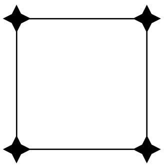
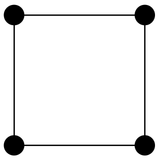
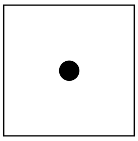
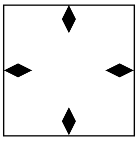
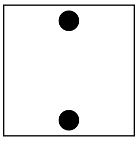
	ρ_d	v	θ_d
V_0	$dQ_0(\Omega)$ 	$RTc^f{}_0(\Omega)$ 	$dW_0(\Omega_H) \otimes Q_1(\Omega_V)$ 
V_1	$dQ_1(\Omega)$ 	$dQ_1^d(\Omega)$ 	$dQ_1(\Omega)$ 
$\widetilde{V}_1$	$Q_1(\Omega)$ 	$Q_1^d(\Omega)$ 	$Q_1(\Omega)$ 
$\widehat{V}_0$	$dQ_0(\Omega)$ 	Broken $RTc^f{}_0(\Omega)$ 	$dQ_0(\Omega_H) \otimes dQ_1(\Omega_V)$ 

Table 4.1: The spaces $\{V_0, V_1, \widetilde{V}_1, \widehat{V}_0\}$ to be used with the recovered advection scheme in this chapter. The domain $\Omega = \Omega_H \otimes \Omega_V$ has an extruded structure and can be in either dimension $d = 2$ or $d = 3$. This figure illustrates the $d = 2$ spaces used for the density ρ_d, velocity v and potential temperature θ_d. V_0 represents the original space of the function, whilst V_1 is the space in which the advection takes place. $\widetilde{V}_1$ the fully-continuous space of same degree as V_1 and $\widehat{V}_0$ is the fully-discontinuous version of V_0. The diagrams represent the locations of the degrees of freedom (DOFs) for each of these spaces, with scalar DOFs represented by circles and vector DOFs represented by diamonds. Although all spaces are tensor products (denoted by $\otimes$) of a space over Ω_H with a space over Ω_V, since the horizontal and vertical degrees are the same for most spaces they are equivalent to another space over Ω. The space $RTc^f{}_k$ is the k-th Raviart-Thomas space. The superscript d represents a space of d-dimensional vector fields.

finite element space can sometimes be equivalent to some other well-known space W on $\Omega = \Omega_H \otimes \Omega_V$. For example $dQ_0(\mathbb{R}) \otimes dQ_0(\mathbb{R}) \equiv dQ_0(\mathbb{R}^2)$. We will use this latter notation for brevity where possible, also omitting the domains.

For each variable the space V_0 is the usual space in which the variable lies. These are the $k = 0$ spaces given in Section 3.2.2 and listed in Table 3.1.

For the advection operator $\mathcal{A}$ to have second-order numerical accuracy, the advection should take place in spaces that are at least linear in each direction. We therefore choose V_1 to be the smallest entirely discontinuous space that is linear in both directions. For ρ_d and θ_d, this is dQ_1, whilst for $\boldsymbol{v}$ this is the vector dQ_1 space, (and so is dQ_1 for each component).

The space $\widetilde{V}_1$ is then formed by taking the completely continuous form of V_1, whilst $\widehat{V}_0$ is formed from the completely broken or discontinuous version of V_0. The full set of spaces is listed in Table 4.1, which also shows the spaces diagrammatically, representing scalar DOFs by circles and vector DOFs by diamonds.

4.2.3 The Recovery Operator

In this section we discuss the details of the recovery operator that we will use, which is based upon the *weighted averaging* operator used in [71]. Our recovery operator reconstructs $q \in V_0$ into $\widetilde{V}_1$ using the following procedure. Let i be a degree of freedom in the space $\widetilde{V}_1$. The value of the field in $\widetilde{V}_1$ at i is determined to be the value of q at the location of i. However as $\widetilde{V}_1$ is continuous, i may be shared between a set of multiple elements $\{e_i\}$. In this case the value in $\widetilde{V}_1$ is the average of the values over $\{e_i\}$.

This operator is described in [71], which implies that it possesses second-order convergence in the L^2-norm when V_0 is the discontinuous constant space dQ_0 and $\widetilde{V}_1$ is Q_1, the space of continuous linear functions over cells. These spaces correspond to those listed in Section 4.2.2 which we will use for our advection schemes. This operator is intended only for use with fields on flat meshes, and must be extended for transport of vector fields on curved meshes.

Error in Recovery Operator

We now briefly demonstrate, for a uniform mesh of orthogonal rectangular elements on an unbounded domain, that this recovery operator does indeed have second-order convergence for $V_0 = dQ_0$ and $\widetilde{V}_1 = Q_1$.

Let Δx be a representative scale of the grid, which is uniform (so that elements are of the same size) and orthogonal at vertices, upon a domain with no boundary. On this mesh we consider the spaces dQ_0, which has DOFs $\{\boldsymbol{x}_j\}$ in the centre of elements, and Q_1 which has DOFs $\{\boldsymbol{x}_i\}$ at the vertices of elements. We consider a smooth function $F(\boldsymbol{x})$ that can be expanded in a Taylor series at every point $\boldsymbol{y} \in \Omega$, such that

$$F(\boldsymbol{x}) = F(\boldsymbol{y}) + (\boldsymbol{x} - \boldsymbol{y}) \cdot \boldsymbol{\nabla} F(\boldsymbol{y}) + \mathcal{O}(\Delta x^2), \tag{4.6}$$

i.e. Δx is much smaller than the radius of curvature of F. If F is projected into $g \in Q_1$ via a Galerkin projection, then as g is locally linear it will represent F at the vertices $\{\boldsymbol{x}_i\}$ to $\mathcal{O}(\Delta x)^2$. Similarly, if F is projected into $f \in dQ_0$ via a Galerkin projection, then at the centres of elements $\{\boldsymbol{x}_j\}$, f will represent F to $\mathcal{O}(\Delta x)^2$ (as taking the Galerkin projection of F into dQ_0 is equivalent to averaging F over the element e_j).

Then f is recovered to $\mathcal{R}f = \widetilde{f} \in Q_1$ via our averaging operator. At a vertex $\boldsymbol{x}_i$, the value of $\widetilde{f}$ is given by

$$\widetilde{f}(\boldsymbol{x}_i) = \frac{1}{N_i}\sum_j f(\boldsymbol{x}_{j,i}), \tag{4.7}$$

where the $\{\boldsymbol{x}_{j,i}\}$ are the DOFs in $\{\boldsymbol{x}_j\}$ of the N_i elements adjacent to $\boldsymbol{x}_i$. Substituting in the values for F and expanding as a Taylor series around $\boldsymbol{x}_i$,

$$\widetilde{f}(\boldsymbol{x}_i) = \frac{1}{N_i}\sum_j^{N_i}\left[F(\boldsymbol{x}_i) + (\boldsymbol{x}_j - \boldsymbol{x}_i)\cdot\boldsymbol{\nabla}F(\boldsymbol{x}_i)\right] + \mathcal{O}(\Delta x^2). \tag{4.8}$$

Since g is accurate to $\mathcal{O}(\Delta x^2)$, then comparing $\widetilde{f}$ with g gives

$$\widetilde{f}(\boldsymbol{x}_i) = g(\boldsymbol{x}_i) + \frac{1}{N_i}\sum_j^{N_i}(\boldsymbol{x}_j - \boldsymbol{x}_i)\cdot\boldsymbol{\nabla}F(\boldsymbol{x}_i) + \mathcal{O}(\Delta x^2). \tag{4.9}$$

As the mesh is uniform and rectangular,

$$\sum_j^{N_i}(\boldsymbol{x}_j - \boldsymbol{x}_i) = 0, \tag{4.10}$$

and so the $\mathcal{O}(\Delta x)$ term will vanish, which demonstrates that the recovery operator is second-order accurate in this case. Under some mild mesh regularity assumptions, the reconstruction operator is second-order accurate for non-uniform meshes. Otherwise, a weighted average may be necessary to ensure second-order convergence.

Boundaries

However, this operation for dQ_0 to Q_1 does not have second-order convergence when representing fields with non-zero gradient at the boundaries of the domain. Whilst the second-order convergence holds for the interior of the domain, at the boundaries DOFs are shared by fewer elements and may not necessarily accurately represent the gradient. If the averaging operator is not to be altered, then another step is required to correct the values of the recovered field at the boundaries. This could involve minimising the curvature within a boundary element subject to the constraints of conservation of mass (from the original field). Alternatively, the inaccurate values at the boundary can be

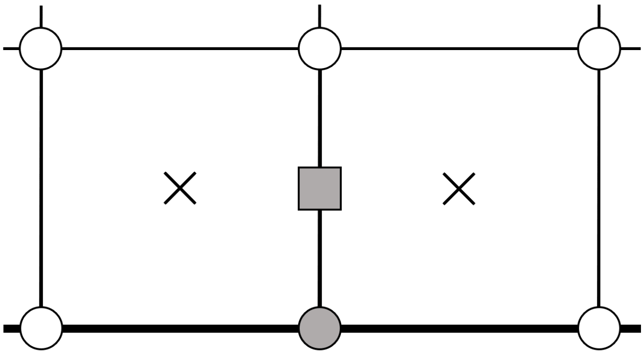


Figure 4.2: The value recovered at the boundary (thick line at the bottom of the figure), when recovering a field in Q_1 from dQ_0, will not accurately represent the true field if it has a gradient at the boundary. This is because it is the average of two values from the interior of cells (shown with crosses), whose average point (shown with a grey square) is not in the same location as the point on the boundary (denoted by a grey circle). The inaccurate value at the grey circle is actually the accurate value the field should have at the square.

exploited through use of a Taylor expansion. This is the premise of the operation that we use, described below.

Procedure for Boundary Recovery

The procedure that we explain here is valid for the recovery of all the spaces laid out in Section 4.2.2 and Table 4.1 for two and three dimensions on flat domains, as well as one-dimensional elements. The values originally recovered at the boundary are inaccurate because the average position of the points being averaged is not the same as the position of the DOF upon the boundary. However, it would be the accurate value of field at the average position of those averaged points. This is illustrated in Figure 4.2. We call this position the effective position. The field is expanded via a Taylor approximation, with unknown coefficients. For instance in two dimensions we obtain the approximation, for each element that

$$q(x, y) = a_0 + a_1x + a_2y + a_3xy. \tag{4.11}$$

Using the accurate interior values and the inaccurate values at their effective positions makes a square linear system of equations that can be solved for the coefficients. These equations are solved element-by-element via Gaussian elimination, and once the coefficients are found, the accurate values on the boundary are obtained by substituting in the coordinates of the values on the boundary. This creates a field in dQ_1, but the final recovered field in Q_1 is restored with another averaging step at each vertex.

The main implementation challenges are to identify which boundary vertices may be inaccurate and to find the corresponding effective positions, which both depend upon the mesh and the original space V_0. Not all boundary vertices will be inaccurate: as shown in Figure 4.3, the original spaces used for θ_d and $\boldsymbol{v}$ can accurately represent gradients in the boundary elements for specific orientations. Rather than dealing which each case

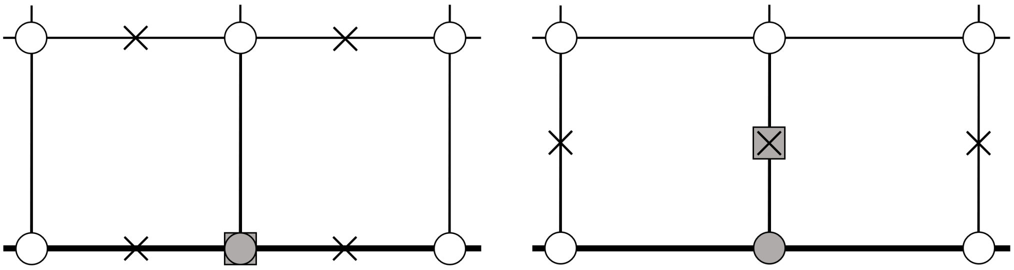


Figure 4.3: The location of effective coordinates for the (left) vertical component of velocity and the (right) horizontal component of velocity for quadrilateral elements along the bottom boundary of a two-dimensional domain. The initial coordinates in the $\mathrm{RTc^f}_0$ space are shown with grey circles, effective coordinates are illustrated with grey squares, whilst original coordinates in Q_1 are shown with crosses. The original coordinates for the boundary vertices of vertical velocity component are the same as the effective coordinates, whilst they are different for the horizontal velocity.

individually[†] we require a general procedure. In all cases, the effective coordinates corresponding to a boundary vertex are halfway between the position of the boundary vertex and its nearest interior vertex. Given a field f in space V_0 that is recovered into $\widetilde{V}_1$, in order to identify which boundary vertices need effective coordinates, we enquire whether vertices are topologically associated with the boundary of the domain.

The whole process encompassed by the recovery operator $\mathcal{R}$ used in Gusto can be summarised as follows:

1. recover from V_0 to $\widetilde{V}_1$ via the averaging operation. Values at boundaries may not be accurate;
2. expand the field as a Taylor series, using the values on the interior and the effective coordinates;
3. solve the resulting linear problem for each cell using Gaussian elimination to obtain the coefficients in the expansion;
4. substitute in the positions of the boundary vertices to obtain accurate values at the boundaries for a field in V_1;
5. perform a final averaging step to obtain the field in $\widetilde{V}_1$.

Boundedness of Recovery Operator

We will now show that the specific recovery operator that we have defined in this section satisfies Assumption 1, i.e. for all $q \in V_0$ there is some $C > 0$ such that $||\mathcal{R}q|| \leqslant C||q||$,

[†] As well as identifying the effective coordinates for the ρ_d and θ_d fields, this procedure is also necessary for each of the components of $\boldsymbol{v}$. This would be necessary in both two and three dimensions, for cases with different boundary conditions (for instance the domain may or not be periodic in one or two horizontal directions). It is desirable for this implementation to also work for set-ups for the shallow-water equations, as well as for vertical slice problems.

where we take $V_0 = dQ_0$ and $\widetilde{V}_1 = Q_1$. This property is used in Section 4.3.1.

Theorem 1. *Consider the action of the specific recovery operator* $\mathcal{R} : V_0 \to \widetilde{V}_1$ *defined in Section 4.2.3 upon a field* $q \in V_0$ *for* $V_0 = dQ_0$ *and* $\widetilde{V}_1 = Q_1$*. There is some* $C > 0$ *such that* $||\mathcal{R}q|| \leqslant C||q||$ *for all* $q \in V_0$*, where* $||\cdot||$ *denotes the* L^2 *norm.*

Proof. We begin by defining $q_{\mathcal{R}} := \mathcal{R}q$. We consider the L^2 norm of $q_{\mathcal{R}}$ over an individual element e_i, $||q_{\mathcal{R}}||^2_{e_i}$. The element e_i and cells that it shares DOFs with form a patch $\mathcal{P}_i$, and we introduce a coordinate scaling $\mathbf{x} \to \mathbf{x}/h$ under which e_i becomes e'_i, which has unit area. If $\{\phi_j(\mathbf{x})\}$ are the M basis functions spanning $\widetilde{V}_1$, then $q_{\mathcal{R}}$ can be written as

$$q_{\mathcal{R}} = \sum_j^M q_{\mathcal{R},j}\phi_j(\mathbf{x}), \tag{4.12}$$

where $q_{\mathcal{R},j}$ is the value of $q_{\mathcal{R}}$ at the j-th DOF. We now consider $||q_{\mathcal{R}}||^2_{e'_i}$

$$||q_{\mathcal{R}}||^2_{e'_i} = \int_{e'_i} \sum_j^M \sum_k^M q_{\mathcal{R},j} q_{\mathcal{R},k} \phi_j(h\mathbf{x})\phi_k(h\mathbf{x})\, dx \equiv ||\mathbf{q}_{\mathcal{R}}||^2_{\Phi',e'_i}, \tag{4.13}$$

where $\mathbf{q}_{\mathcal{R}}$ is the vector of values of $q_{\mathcal{R}}$ at DOFs and $||\cdot||_{\Phi'} \equiv ||\mathbf{y}^T\Phi'\mathbf{y}||$ denotes the norm with the mass matrix $\Phi' := \int \phi_j(h\mathbf{x})\phi_k(h\mathbf{x})\, dx$ acting upon some M-dimensional vector $\mathbf{y}$. From norm equivalence, we know that for some $C > 0$, $\mathbf{q}_{\mathcal{R}}$ evaluated in the Φ' norm can be bounded from above by evaluation with the vector norm, and we can write this as the sum of components

$$||q_{\mathcal{R}}||^2_{e'_i} \leqslant C_i \sum_{j \in e'_i}^M \left(q_{\mathcal{R},j}\right)^2, \tag{4.14}$$

where C_i is a constant that depends upon the size of the element. If the j-th DOF in $\widetilde{V}_1$ is shared between N_j cells, then $q_{\mathcal{R},j}$ is the average values of q in those cells, and hence $q_{\mathcal{R},j} = \frac{1}{N_j}\sum_k^{N_j} q_k$, giving

$$\begin{aligned} ||q_{\mathcal{R}}||^2_{e'_i} \leqslant C_i \sum_{j \in e'_i}^M \left(\frac{1}{N_j}\sum_k^{N_j} q_k\right)^2 &\leqslant C_i \sum_{j \in e'_i}^M \left(\sum_k^{N_j} q_k\right)^2 \\ &\leqslant C_i \sum_{j \in e'_i}^M \sum_k^{N_j} (q_k)^2 \leqslant C_i ||q||^2_{\mathcal{P}'_i}, \end{aligned} \tag{4.15}$$

where the equalities follow as N_j is a positive integer, from the Cauchy-Schwarz inequality and from the definition of the L^2 norm in V_0. The constant C_i has absorbed the effect of double-counting of cells over the patch. Under some regularity assumptions about the shape of the mesh,

$$||q_{\mathcal{R}}||^2_{e_i} \leqslant C^*||q||^2_{\mathcal{P}_i} \tag{4.16}$$

where C^* is the maximum of C_i over the mesh. Now, considering the L^2 norm over the whole domain,

$$||q_{\mathcal{R}}||^2 \leqslant \sum_i ||q_{\mathcal{R}}||^2_{e_i} \leqslant \sum_i C_i ||q||^2_{\mathcal{P}_i} \leqslant C||q||^2, \tag{4.17}$$

where the constant again takes double-counting into account. This also holds for the procedure at the boundaries, where the values of the reconstructed field are a linear function of the interior and original field values. Thus we arrive at the conclusion that for some $C > 0$,

$$||q_{\mathcal{R}}|| \leqslant C||q||. \tag{4.18}$$

□

4.2.4 Limiting

In numerical weather or climate models, there may be many additional prognostic variables representing moisture or chemical species. These variables will typically lie in either the same space as the density ρ_d or the potential temperature θ_d. In this thesis we will consider only moisture variables, which are the focus of Chapter 5. These will lie in the space of θ_d, which can simplify the thermodynamics associated with phase changes. However, it may come at the cost of sacrificing conservation of the mass of water, although this could be remedied by solving the transport equation in *conservative* form (4.2) for $r\rho_d$ rather than in *advective* form for tracer r.

The continuous equations describing the advection of these tracer variables have monotonicity and shape-preserving properties; however the discrete representation may not replicate these properties, which may lead to unphysical solutions such as negative concentrations. This can be avoided by the application of slope limiters.

In the recovered scheme, both the advection operator $\mathcal{A}$ and final projection operator $\mathcal{P}$ may produce spurious overshoots and undershoots, and so need limiting. In the case of the projection operator, we do this by using the second projection operator $\mathcal{P}_B$. This prevents the formation of new maxima and minima as it is composed of two bounded operations: the projection into the broken lower-order space and then the recovery of continuity. For the set of spaces proposed in Section 4.2.2, to limit the advection operator $\mathcal{A}$, we use the vertex-based limiter outlined in [75]. This limiter divides the field in each element into a constant mean part and a linear perturbation. Considering the values of neighbouring elements at shared vertices gives upper and lower bounds. The limited field is then the mean part plus a constant times the perturbation, so that the field remains bounded. This limiter is applied to the field before the advection operator begins, and after each stage of $\mathcal{A}$.

4.3 Properties of the Numerical Scheme

4.3.1 Stability

Here we will show the stability of the 'recovered space' scheme, following a similar argument to that used in [67]. First, we will need the following Lemma.

Lemma 1. *Let the operator $\mathcal{J} : V_0 \to V_1$ be defined by*

$$\mathcal{J} := \mathcal{I}(\mathcal{R} - \widehat{\mathcal{P}}\mathcal{R} + 1), \tag{4.19}$$

so that the 'recovered space' scheme can be written as

$$q^{n+1} = \mathcal{P}\mathcal{A}\mathcal{J}q^n. \tag{4.20}$$

Denote by $||\cdot||$ the L^2 norm. Then $||\mathcal{J}q|| \leqslant \kappa||q||$ for some $\kappa > 0$ for all $q \in V_0$.

Proof. From the definition of $\mathcal{J}$,

$$||\mathcal{J}q|| = ||q + \mathcal{R}q - \widehat{\mathcal{P}}\mathcal{R}q||. \tag{4.21}$$

By applying the triangle inequality,

$$||\mathcal{J}q|| \leqslant ||q|| + ||\mathcal{R}q|| + ||\widehat{\mathcal{P}}\mathcal{R}q||. \tag{4.22}$$

We will now inspect the $||\widehat{\mathcal{P}}\mathcal{R}q||$ term. The definition of $\widehat{\mathcal{P}}$ is that $\int_\Omega \widehat{\psi}\widetilde{q}\,dx = \int_\Omega \widehat{\psi}\widehat{\mathcal{P}}\widetilde{q}\,dx$ for all $\widehat{\psi} \in \widehat{V}_0$, where $\widetilde{q} \in \widetilde{V}$. Since $\widehat{\mathcal{P}}\widetilde{q} \in \widehat{V}_0$, then it follows that

$$\int_\Omega (\widehat{\mathcal{P}}\widetilde{q})^2\,dx = \int_\Omega (\widehat{\mathcal{P}}\widetilde{q})\widetilde{q}\,dx. \tag{4.23}$$

Now we consider the integral $\int_\Omega (\widehat{\mathcal{P}}\widetilde{q} - \widetilde{q})^2\,dx$, which cannot be negative. Expanding this integral and using the result (4.23) gives $\int_\Omega \widetilde{q}^2\,dx \geqslant \int_\Omega (\widehat{\mathcal{P}}\widetilde{q})^2\,dx$, in other words that $||\widehat{\mathcal{P}}\mathcal{R}q|| \leqslant ||\mathcal{R}q||$. Hence, returning to $||\mathcal{J}q||$, we obtain

$$||\mathcal{J}q|| \leqslant ||q|| + 2||\mathcal{R}q||. \tag{4.24}$$

Finally, we use the property of $\mathcal{R}$ that $||\mathcal{R}q|| \leqslant C||q||$ for some $C > 0$, and so letting $\kappa = 1 + 2C$ we arrive at

$$||\mathcal{J}q|| \leqslant \kappa||q||. \tag{4.25}$$

This completes the proof. □

Theorem 2. *Let the advection operator $\mathcal{A}$ have a stability constant α, such that*

$$||\mathcal{A}|| := \sup_{p \in V_1, ||p||>0} \frac{||\mathcal{A}p||}{||p||} \leqslant \alpha. \tag{4.26}$$

Then the stability constant α^* of the 'recovered space' scheme on V_0 satisfies $\alpha^* = \kappa\alpha$ for some constant κ.

Proof. Since from Lemma 1, $||\mathcal{J}q|| \leqslant \kappa||q||$ for some $\kappa > 1$,

$$\sup_{q \in V_0, ||q||>0} \frac{||\mathcal{P}\mathcal{A}\mathcal{J}q||}{||q||} \leqslant \sup_{q \in V_0, ||q||>0} \kappa \frac{||\mathcal{P}\mathcal{A}\mathcal{J}q||}{||\mathcal{J}q||}. \tag{4.27}$$

As $V_0 \subset V_1$, the supremum over elements in V_1 cannot be smaller than the supremum over elements in V_0. Recognising that $\mathcal{J}q \in V_1$,

$$\sup_{q \in V_0, ||q||>0} \frac{||\mathcal{P}\mathcal{A}\mathcal{J}q||}{||\mathcal{J}q||} \leqslant \sup_{p \in V_1, ||p||>0} \frac{||\mathcal{P}\mathcal{A}p||}{||p||}. \tag{4.28}$$

For the final step, we must consider both cases $\mathcal{P}_A$ and $\mathcal{P}_B$ for the projection operator. In the case of $\mathcal{P}_A$, we can use a similar argument to that of the projection operator in Lemma 1 to obtain that $||\mathcal{P}\mathcal{A}p|| \leqslant ||\mathcal{A}p||$. For $\mathcal{P}_B = \mathcal{P}_R\mathcal{P}_I$, each step maps a function into a space that is smaller; i.e. $\widehat{V}_0 \subset V_1$ and $V_0 \subset \widehat{V}_0$, so that the supremum of $||\mathcal{P}\mathcal{A}p||$ must be smaller than the supremum of $||\mathcal{A}p||$. In both cases we obtain that

$$\sup_{p \in V_1, ||p||>0} \frac{||\mathcal{P}\mathcal{A}p||}{||p||} \leqslant \sup_{p \in V_1, ||p||>0} \frac{||\mathcal{A}p||}{||p||} \leqslant \alpha, \tag{4.29}$$

where the final inequality defines the stability of $\mathcal{A}$. Combining these arguments together gives

$$\sup_{q \in V_0, ||q||>0} \frac{||\mathcal{P}\mathcal{A}\mathcal{J}q||}{||q||} \leqslant \kappa\alpha, \tag{4.30}$$

and so the stability constant of the 'recovered space' scheme is $\alpha^* = \kappa\alpha$. □

4.3.2 Von Neumann Analysis

Now we will attempt to identify the stability constant for three one-dimensional examples, by performing Von Neumann stability analysis. This can also be used with the Courant-Friedrich-Lewy (CFL) condition to give upper limits to stable Courant numbers.

The three cases that will be considered are that of $V_0 = dQ_0$ (which might represent the advection of ρ_d), and the two cases of $V_0 = Q_1$ with $\mathcal{P}_A$ and $\mathcal{P}_B$ as the projection operators (for advection of velocity and moisture respectively). The same advection op-

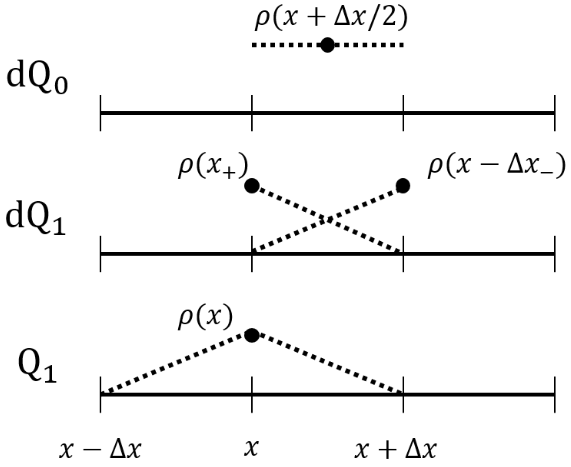


Figure 4.4: The degrees of freedom and basis functions for each of the spaces used in the Von Neumann analysis.

erator discretising the advective form (4.1) of the transport equation will be used for all three cases, with the advection taking place in $V_1 = dQ_1$.

In each case we consider a periodic domain of length L, divided into N cells, each of length Δx. We will make the assumption that our function $q_n(x)$ at the n-th time step can be written as a sum of Fourier modes,

$$q^n(x) = \sum_k A_k^n e^{ikx}. \tag{4.31}$$

Then for the k-th mode $q_k^n(x+\Delta x) = q_k^n(x)e^{ik\Delta x}$.

The basis functions of the three spaces relevant to this analysis, dQ$_0$, Q$_1$ and dQ$_1$, are shown in Figure 4.4.

Advection Operator

First we described the advection operator $\mathcal{A}_k$ acting upon the k-th mode of a function in dQ$_1$. In each cell, the function can be described by two components: evaluation of the field at each of the two cell boundaries. For the advection, we use the simple upwinding scheme with a forward Euler time discretisation, within the framework of a three-step Runge-Kutta scheme, as outlined in Section 2.2.2. We describe the action of a single forward Euler step with the operator $\mathcal{L}_k$. This is determined by discretising the one-dimensional advection equation with constant $\nu > 0$,

$$\frac{\partial q}{\partial t} + \nu\frac{\partial q}{\partial x} = 0, \tag{4.32}$$

for $q \in V_1$ by integrating with the test function $\psi \in V_1$. Taking equation (2.12) from Section 2.2.2 in one dimension, with constant velocity $v > 0$, we obtain

$$\int_\Omega \psi \mathcal{L} q^n \, dx = -v\Delta t \int_\Gamma [\![\psi]\!]_+ \, q^\dagger \, dS + v\Delta t \int_\Omega q^n \frac{\partial \psi}{\partial x} \, dx. \tag{4.33}$$

Making the assumption that $q(x) = q(x + \Delta x)e^{-ik\Delta x}$, and using that q is piecewise linear, we can write down a representation of $\mathcal{L}_k$ for the DOFs on either side of a given cell:

$$\mathcal{L}_k = \begin{pmatrix} -3c & 4ce^{-ik\Delta x} - c \\ 3c & -c - 2ce^{-ik\Delta x} \end{pmatrix}. \tag{4.34}$$

We then obtain the full advection operator by using the three-step Runge-Kutta scheme of (2.13) outlined in [28], so that the overall operator is

$$\mathcal{A}_k = \mathbb{1} + \mathcal{L}_k + \frac{1}{2}\mathcal{L}_k^2 + \frac{1}{6}\mathcal{L}_k^3, \tag{4.35}$$

where $\mathbb{1}$ is the identity operator. We omit the matrix representation of $\mathcal{A}_k$ here for brevity.

Case A: dQ_0

This represents the advection of density ρ_d, or the velocity $\boldsymbol{v}$ perpendicular to its direction. The set of spaces $\{V_0, V_1, \widetilde{V}_1, \widehat{V}_0\}$ is $\{dQ_0, dQ_1, Q_1, dQ_0\}$.

Then, for a given cell and Fourier mode, the operators can be represented in the following matrix forms:

$$\widehat{\mathcal{P}} = \mathcal{P} = \frac{1}{2}\begin{pmatrix} 1 & 1 \end{pmatrix}, \quad \mathcal{R}_k = \frac{1}{2}\begin{pmatrix} 1 + e^{-ik\Delta x} \\ e^{ik\Delta x} + 1 \end{pmatrix}, \quad \mathcal{I} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}. \tag{4.36}$$

Combining these operators, the advection scheme for the k-th mode of $q^n(x)$ is then expressed as

$$q_k^{n+1} = \frac{1}{4}\begin{pmatrix} 1 & 1 \end{pmatrix} \mathcal{A}_k \begin{pmatrix} 2 - i\sin(k\Delta x) \\ 2 + i\sin(k\Delta x) \end{pmatrix} q_k^n. \tag{4.37}$$

Following the analysis through and writing $\phi = k\Delta x$ and $c = v\Delta t/\Delta x$ gives a stability constant

$$\begin{aligned} |A_k|^2 = c^2 \Bigg[& c^3 \left(\frac{13}{4}\cos\phi - \frac{5}{3}\cos 2\phi + \frac{1}{12}\cos 3\phi + \frac{1}{6}\cos 4\phi - \frac{11}{6} \right) \\ & + c^2 \left(\frac{29}{12}\sin\phi - \frac{5}{3}\sin 2\phi + \frac{1}{12}\sin 3\phi + \frac{1}{6}\sin 4\phi - \frac{7}{4}\cos\phi + \frac{3}{4}\cos 2\phi - \frac{1}{4}\cos 3\phi + \frac{5}{4} \right) \\ & + c \left(-\frac{3}{4}\sin\phi + \frac{3}{4}\sin 2\phi - \frac{1}{4}\sin 3\phi - \cos\phi + \frac{1}{4}\cos 2\phi + \frac{3}{4} \right) - \frac{3}{2}\sin\phi + \frac{1}{4}\sin^2 2\phi - 1 \Bigg]^2. \end{aligned} \tag{4.38}$$

CASE B: Q_1 WITH $\mathcal{P} = \mathcal{P}_A$

In this case, the set of spaces $\{V_0, V_1, \widetilde{V}_1, \widehat{V}_0\}$ is $\{Q_1, dQ_1, Q_1, dQ_1\}$. This describes advection of velocity parallel to its direction, or of potential temperature without bounding the final projection step. The operators can be represented by

$$\begin{aligned} \mathcal{P}_A &= \frac{1}{4 + 2\cos k\Delta x}\begin{pmatrix} 2 + e^{-ik\Delta x} & 1 + 2e^{-ik\Delta x} \end{pmatrix}, \\ \widehat{\mathcal{P}} &= \mathcal{I} = \begin{pmatrix} 1 \\ e^{ik\Delta x} \end{pmatrix}, \end{aligned} \tag{4.39}$$

where the projection operator $\mathcal{P}_A$ has been determined by solving equation (4.4). The recovery operator is the identity, and since the injection $\mathcal{I}$ and the projection $\widehat{\mathcal{P}}$ are equivalent in this case the scheme acting upon q^n becomes $q^{n+1} = \mathcal{P}_A \mathcal{A} \mathcal{I} q^n$. Following through the analysis gives

$$\begin{aligned} |A_k|^2 = \left(\frac{c}{\cos\phi + 2}\right)^2 \Bigg[& \left(c^2\cos\phi - c^2 + 3\right)^2 \sin^2\phi \\ & + \left(-2c^3\cos\phi + \frac{1}{2}c^3\cos 2\phi + \frac{3}{2}c^3 \quad + 3c^2\cos\phi - 3c^2 + \cos\phi + 2\right)^2 \Bigg]. \end{aligned} \tag{4.40}$$

CASE C: Q_1 WITH $\mathcal{P} = \mathcal{P}_B$

For this case, the set of spaces are the same as in the second case. The only difference is that the projection operator $\mathcal{P}$ is now $\mathcal{P}_B = \mathcal{P}_R\mathcal{P}_I$. As $V_1 = \widehat{V}_0 = dQ_1$, the interpolation $\mathcal{P}_I$ is the identity, and $\mathcal{P}_B = P_R$. The operators are

$$\mathcal{P}_B = \frac{1}{2}\begin{pmatrix} 1 & e^{-ik\Delta x} \end{pmatrix}, \quad \widehat{\mathcal{P}} = \mathcal{I} = \begin{pmatrix} 1 \\ e^{ik\Delta x} \end{pmatrix}, \tag{4.41}$$

which gives leads to the amplification factor

$$\begin{aligned} |A_k|^2 = & \left(\frac{2}{3}c^3\sin\phi + \frac{1}{6}c^3\sin 2\phi - \frac{1}{3}c^3\sin 3\phi - c^2\sin\phi + \frac{1}{2}c^2\sin 2\phi - c\sin\phi\right)^2 \\ + & \left(-\frac{7}{3}c^3\cos\phi - \frac{1}{6}c^3\cos 2\phi + \frac{1}{3}c^3\cos 3\phi + \frac{13}{6}c^3 + 3c^2\cos\phi - \frac{1}{2}c^2\cos 2\phi - \frac{5}{2}c^2 + 1\right)^2. \end{aligned} \tag{4.42}$$

4.3.3 Critical Courant Numbers

The Courant-Friedrich-Lewy (CFL) criterion says that an advection scheme with amplification factor $|A_k| > 1$ may not be stable. The critical Courant number c^* is the lowest Courant number $c = v\Delta t/\Delta x$ such that the amplification factor is greater than unity. We numerically measured the critical Courant numbers for the three cases laid out in Section 4.3.2, and these are displayed in Table 4.2. Although case C has a significantly lower critical Courant number, the intention is to run this scheme with a limiter or with a subcycling time discretisation, allowing it to be used at higher Courant numbers. Instances of work using these kind of limiters are [67] and [75].

Examples of critical Courant numbers for other upwinding schemes can be found in Table 2.2 of [27]. The most relevant comparison that can be made from this is to that of polynomials of degree 1 with a Runge-Kutta method of order 3, which has a critical Courant number of 0.409. A space of discontinuous linear polynomials has the same number of degrees of freedom as a space of discontinuous constants but with half the grid size, and thus improvements are made if the critical Courant number is more than twice that of the transport scheme for the linear functions. We do therefore observe that the critical Courant numbers for cases A and B are improvements on the discontinuous upwinding scheme applied just to discontinuous linear functions.

Case	A	B	C
c^*	0.8506	0.9930	0.3625

Table 4.2: The critical Courant numbers for the three cases of the advection scheme outlined in Section 4.3.2.

4.4 Numerical Tests

4.4.1 Numerical Accuracy

To verify the numerical accuracy of the scheme, we performed a series of convergence tests. The aim is to find how the error due to advection changes with the grid spacing Δx. We used tests that have an analytic solution in the limit that $\Delta x \to 0$, and compare the final advected profile q with the 'true' profile q_h, which is the analytic solution projected into the relevant function space. This gives an error $||q - q_h||$ (where $||\cdot||$ denotes the L^2 norm) which is calculated for the same problem at different resolutions, and the errors are plotted as a function of the grid spacing Δx. The order of the numerical accuracy is the number n such that $||q - q_h|| \sim \mathcal{O}(\Delta x^n)$, which can measured from the slope of a plot of $\ln(||q - q_h||)$ against $\ln(\Delta x)$. For simplicity, the tests we used are designed so that the 'true' profile is the same as the initial condition.

The initial conditions were obtained by pointwise evaluation of the expressions into higher order fields (we used Q_3). These were then projected into the correct fields. The advecting velocity used lay in the RT_1 space. To mimic how the scheme might be used in a numerical weather model, we performed some of the tests on the different sets of spaces laid out in Section 4.2.2 and the configurations described in Section 4.3.2. Each set of spaces is labelled by the variable name in Table 4.1, with $\left\{V_0, V_1, \widetilde{V}_1, \widehat{V}_0\right\}$ for the fields ρ_d, $\boldsymbol{v}$, θ_d and r given by

$$\rho_d \in \{dQ_0,\ dQ_1,\ Q_1,\ dQ_0\}, \tag{4.43}$$

$$\boldsymbol{v} \in \left\{RT_0,\ dQ_1^d,\ Q_1^d,\ \text{broken}\ RT_0\right\}, \tag{4.44}$$

$$\theta_d \in \{dQ_0 \otimes Q_1,\ dQ_1,\ Q_1,\ dQ_0 \otimes dQ_1\}, \tag{4.45}$$

$$r \in \{dQ_0 \otimes Q_1,\ dQ_1,\ Q_1,\ dQ_0 \otimes dQ_1\}, \tag{4.46}$$

where the superscript d recognises that the space has vector valued nodes. While the scheme labelled θ_d uses the projection operator $\mathcal{P}_A$, the scheme labelled r represents a moisture variable, so uses the same spaces as θ_d but the projection operator $\mathcal{P}_B$ and the vertex-based limiter of [75]. All tests solve the transport equation in the *advective* form.

The first three tests involve advection around a 2D domain representing a vertical slice that is periodic along its side edges but has rigid walls at the top and bottom. The final test is performed over the surface of a sphere. All the vertical slice tests use a domain of height and width 1 m and advect the profile with time steps of $\Delta t = 10^{-4}$ s for a total time of $T = 1$ s.

Rotational Convergence Test

The first test involves a rigid body rotation of a Gaussian profile around the centre of the domain. Using x and z as the horizontal and vertical coordinates, defining $r^2 = (x - x_0)^2 + (z - z_0)^2$ for $x_0 = 0.375$ m, $z_0 = 0.5$ m and using $r_0 = 1/8$ m, the initial condition used for all fields was

$$q = e^{-(r/r_0)^2}. \tag{4.47}$$

For the velocity variable, this initial profile was used for each component of the field. The advecting velocity is generated from a stream function ψ via $\boldsymbol{v} = (-\partial_z\psi, \partial_x\psi)$. Defining $r_v^2 = (x - x_v)^2 + (z - z_v)^2$ with $x_v = 0.5$ m and $z_v = 0.5$ m, the stream function used was:

$$\psi(\boldsymbol{x}) = \begin{cases} \pi(r_v^2 - 0.5), & r_v < r_1, \\ Ar_v^2 + Br_v + C, & r_1 \leqslant r_v < r_2, \\ Ar_2^2 + Br_2 + C, & r_v \geqslant r_2. \end{cases} \tag{4.48}$$

This is designed to be a rigid body rotation for $r_v < r_1$, with no velocity for $r_v \geqslant r_2$ to prevent spurious noise being generated from the edge of the domain. The stream function and its derivative vary smoothly for $r_1 \leqslant r_\nu < r_2$. We use $r_1 = 0.48$ m and $r_2 = 0.5$ m, with $A = \pi r_1/(r_1 - r_2)$, $B = -2Ar_2$ and $C = \pi(r_1^2 - 0.5) - Ar_1^2 - Br_1$. Results showing second order numerical accuracy can be found in Figure 4.7 (left). Initial and final fields for the density in the lowest resolution run ($\Delta x = 0.01$ m) are displayed in Figure 4.5.

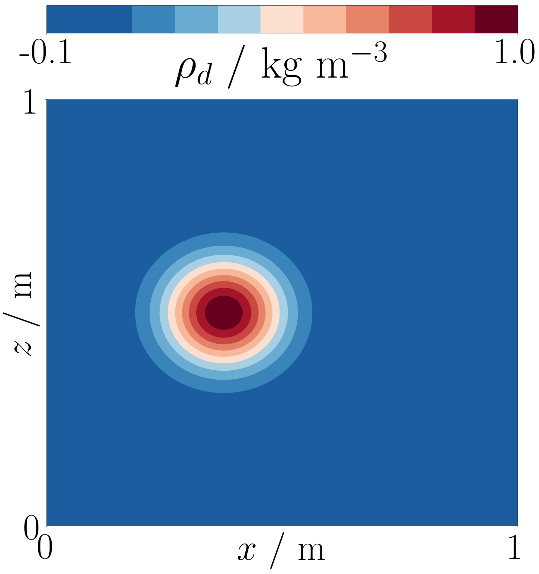


Figure 4.5: The initial (left) and final (right) fields in the dQ_0 space from the rotational convergence test of Section 4.4.1, showing the field labelled ρ_d in Figure 4.7 (left) at the lowest resolution ($\Delta x = 0.01$ m). The field is contoured every 0.1 kg m^{-3} and the contour at 0 kg m^{-3} has been removed for clarity. This convergence test is the simplest that we used to investigate the recovered advection scheme, performing a solid body rotation around the centre of the domain. Almost no difference is visible in the main profile.

Deformational Convergence Test

The second test is a more challenging convergence test, based on the deformational flow experiment described in [67]. The initial profiles were the same as used in the rotational advection test, but with $x_0 = 0.5$ m. The advecting velocity was that of [67]:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \begin{pmatrix} 1 - 5(0.5 - t)\sin(2\pi(x - t))\cos(\pi z) \\ 5(0.5 - t)\cos(2\pi(x - t))\sin(\pi z) \end{pmatrix}. \tag{4.49}$$

Figure 4.7 (right) plots the results of this test, with each variable measuring second order numerical accuracy. Initial and final fields for the density in the lowest resolution run ($\Delta x = 0.01$ m) are displayed in Figure 4.6.

Boundary Convergence Test

The third test was intended to investigate the integrity of the advection scheme at the boundaries of the domain. We use the following reversible flow, which squashes the

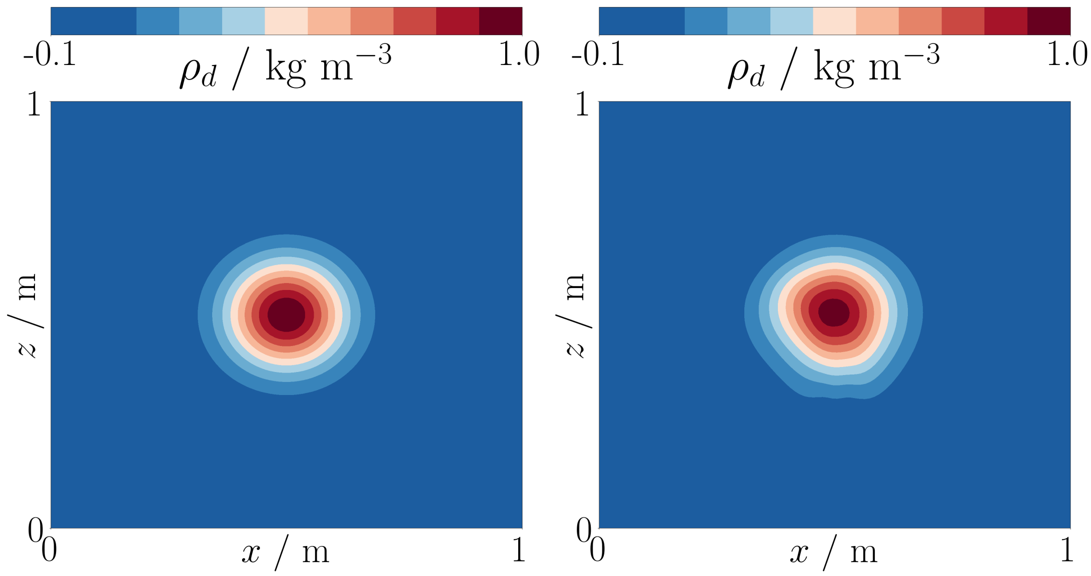


Figure 4.6: The initial (left) and final (right) fields in the dQ_0 space from the deformational convergence test of Section 4.4.1, showing the field labelled ρ_d in Figure 4.7 (right) at the lowest resolution ($\Delta x = 0.01$ m). The field is contoured every 0.1 kg m^{-3} and the contour at 0 kg m^{-3} has been removed for clarity. This convergence test is more challenging, as it involves a time-varying flow which deforms the advected profile.

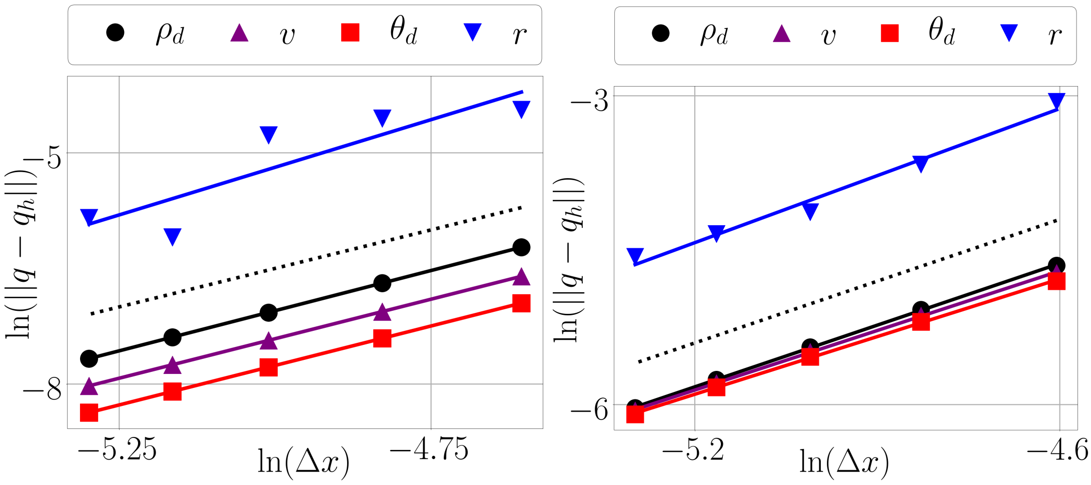


Figure 4.7: Results from convergence tests for the recovered space scheme, plotting, as a function of grid spacing Δx, the error in an advected solution q against the true solution q_h. The different lines labelled ρ_d, v and θ_d represent performing the test in each of the different sets of spaces laid out in Section 4.2.2 and using the projection operator $\mathcal{P}_A$, whilst the line labelled r represents advection with the same spaces as θ_d, but with the projection operator $\mathcal{P}_B$. The dotted line represents an error proportional to Δx^2. (Left) The test describes a rigid body rotation. In all cases, the slopes are around 2, indicating second order numerical accuracy. (Right) A more difficult convergence test featuring deformational flow. Accuracy is approaching second order for each case.

advected material into the boundary before recovering it:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \begin{cases} (1, -\sin(2\pi z)) & t < 0.5 \\ (1, \sin(2\pi z)) & t \geqslant 0.5 \end{cases}. \tag{4.50}$$

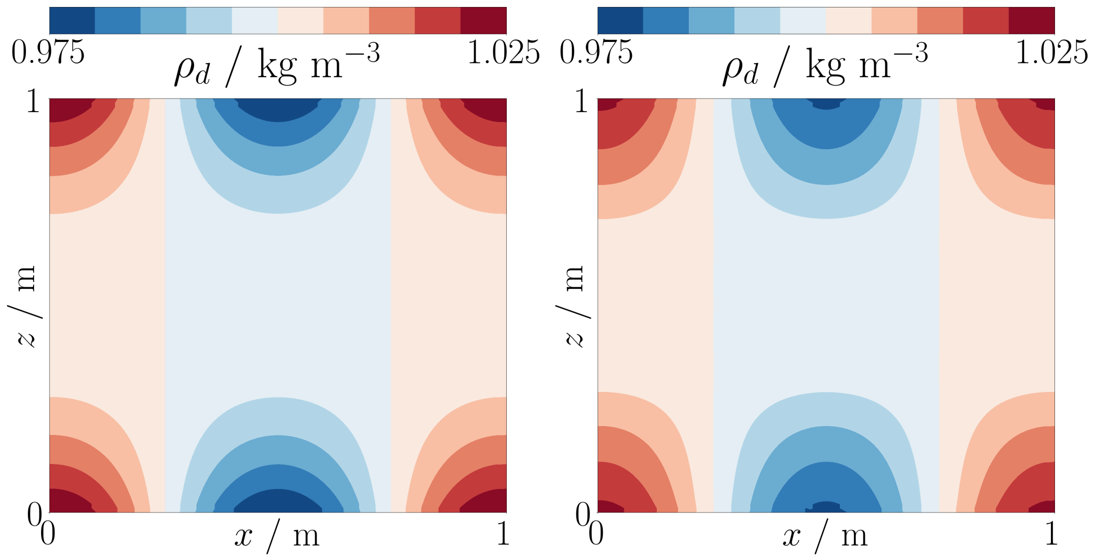


Figure 4.8: Initial (left) and final (right) fields in the dQ_0 space of the boundary convergence test of Section 4.4.1, showing the field labelled ρ in Figure 4.10 (left) at the lowest resolution ($\Delta x = 0.01$ m). The field is contoured every 0.005 kg m^{-3}. This test involves flow that compresses the advected profile into the domain boundary, before reversing it. The test is designed to investigate the efficacy of the extra step that recovers the advected field at boundary. The advection scheme can then be compared with and without this extra step.

The initial condition was

$$q = 1 + \frac{1}{10}\left(z - \frac{1}{2}\right)^2 \cos(2\pi x). \tag{4.51}$$

To see the effect of the extra recovery performed at the boundary described in Section 4.2.3, this extra recovery was turned off for the variables labelled with an asterisk. The results in Figure 4.10 (left) demonstrate that without doing extra recovery at the boundaries, the whole recovery process does not have second order numerical accuracy. Initial and final fields for the density in the lowest resolution run ($\Delta x = 0.01$ m) are displayed in Figure 4.8.

Spherical Convergence Test

The final convergence test was performed on the surface of a sphere, investigating how this transport scheme might perform in a global atmospheric or oceanic model. In this case we used a cubed sphere mesh of a sphere of radius 100 m. The advecting velocity field used was $\boldsymbol{v} = U \sin\lambda$, for latitude λ and $U = \pi/10$ m s^{-1}, which gave a constant zonal rotation rate about the sphere. We took time steps of $\Delta t = 0.5$ s up to a total time of 2000 s so that the initial profile should be equal to the 'true' profile. The initial profile that we used was very similar to that used in the first test case of [76]:

$$q = \begin{cases} \frac{1}{2}\left[1 + \cos\left(\frac{\pi r}{R}\right)\right], & r < R, \\ 0, & \text{otherwise}, \end{cases} \tag{4.52}$$

where $R = 100/3$ m and for latitude λ and longitude φ with $\lambda_0 = 0$ and $\varphi_0 = -\pi/2$, and where r is now given by

$$r = 100 \text{ m} \times \cos^{-1}\left[\sin\lambda_0 \sin\lambda + \cos\lambda_0 \cos\lambda \cos(\varphi - \varphi_0)\right]. \tag{4.53}$$

The errors of this test as a function of resolution are plotted in Figure 4.10 (right). This also appears to show second order accuracy. We found that at lower resolutions, the errors due to the advective scheme were obscured by those from the imperfect discretisation of the surface of the sphere. The initial and final fields of this test are plotted for the coarsest resolution in Figure 4.9.

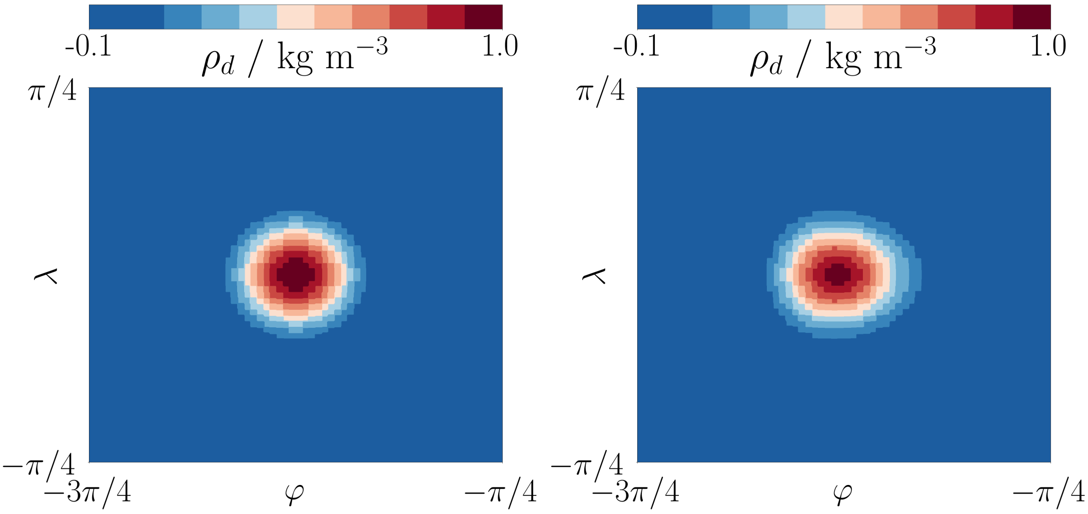


Figure 4.9: The initial (left) and final (right) fields in the dQ_0 space from the spherical convergence test of Section 4.4.1, showing the field labelled ρ in Figure 4.10 (right) at the lowest resolution (the sixth refinement level of the cubed sphere, with $\Delta x \approx 1.6$ m). The axes are longitude φ and latitude λ. The field is contoured every 0.1 kg m^{-3} and the contour at 0 kg m^{-3} has been removed for clarity. This tests the advection over the sphere of a scalar profile, but the scheme needs extending to accurately describe transport of vector-valued fields. We found this test to be difficult to configure and needed the mesh to use a high-order (3 or more) approximation to the sphere in order to obtain reasonable results. The pixellation visible in these images relates to the mapping from the spherical surface to a Cartesian one.

4.4.2 Stability

We tested the formulae (4.38), (4.40) and (4.42) by advecting sine and cosine waves for each of the cases defined. The domain used was a square vertical slice of length 120 m with grid spacing $\Delta x = 1$ m. The amplification factor for a given wavenumber k and Courant number c was measured by advecting a sine and cosine wave of wavenumber k by a constant horizontal velocity c for a single time step of $\Delta t = 1$ s. As before, the domain had periodic boundary conditions on the vertical walls. The amplification factor was then found by measuring the amplitude of the sine and cosine components after the first time step. This was done for several values of c.

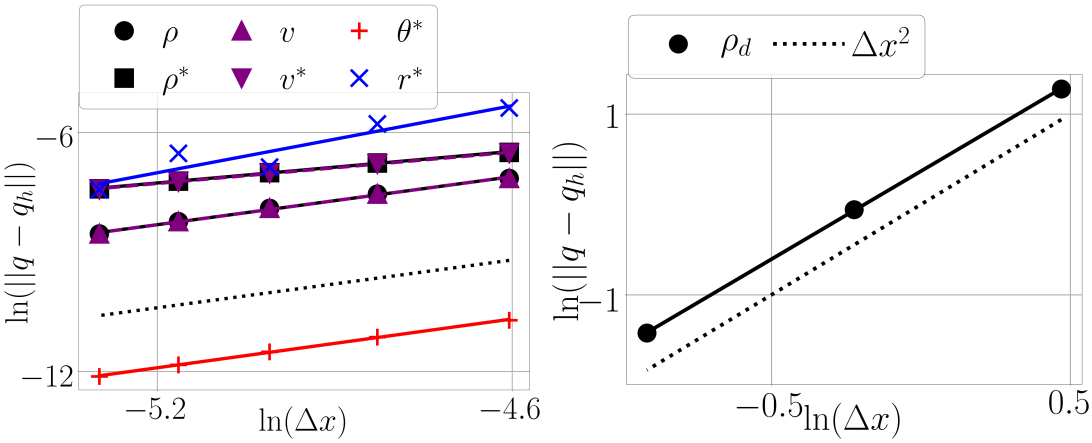


Figure 4.10: More results from convergence tests for the recovered space scheme, plotting, as a function of grid spacing Δx, the error in an advected solution q against the true solution q_h. The dotted line represents an error falling with Δx^2. (Left) A test demonstrating the need for the extra recovery at the boundaries, by comparing the scheme with and without this extra recovery process. The schemes without extra recovery at the boundaries are denoted with an asterisk. They not only display a larger error, but also show lower accuracy. As θ_d and r are linear in the vertical, they are accurately represented at the boundary by the recovery scheme without performing any additional recovery at the boundary. However if rigid walls were present on the side of the domain, these fields would require additional recovery at these boundaries. (Right) The test performed on a cubed sphere mesh. The slope here is very close to 2, again supporting the claim that the advection scheme has second order numerical accuracy.

The measured values are compared with those from the formula in Figure 4.11, which shows agreement for each of the cases considered in Section 4.3.2.

4.4.3 Limiting

The efficacy of the limiting scheme was tested by using the slotted-cylinder, hump, cone set-up originally defined in [77] and used in both [75] and [67]. The advected field was initialised with this condition, lying in the $dQ_0 \otimes Q_1$ space to mimic moisture variables, before a solid-body rotation was completed. This was performed for the bounded case of the scheme defined in Section 4.2.1, using the projection operator $\mathcal{P}_B$ and the vertex-based limiter of [75] for the advection. The resulting field is shown in Figure 4.12 where it is also compared to the rotation of a field in the Q_1 space, using the same limited advection scheme, but without the 'recovered' parts of the scheme. The field does indeed remain bounded, suggesting that the limiter has worked well.

4.5 Recovered Diffusion

One simple technique for parametrising turbulent flows or for ensuring stability in atmospheric models is to add artificial diffusion. We will consider it added to the temperature and velocity equations, as shown in equations (3.1). The diffusion equation that we will

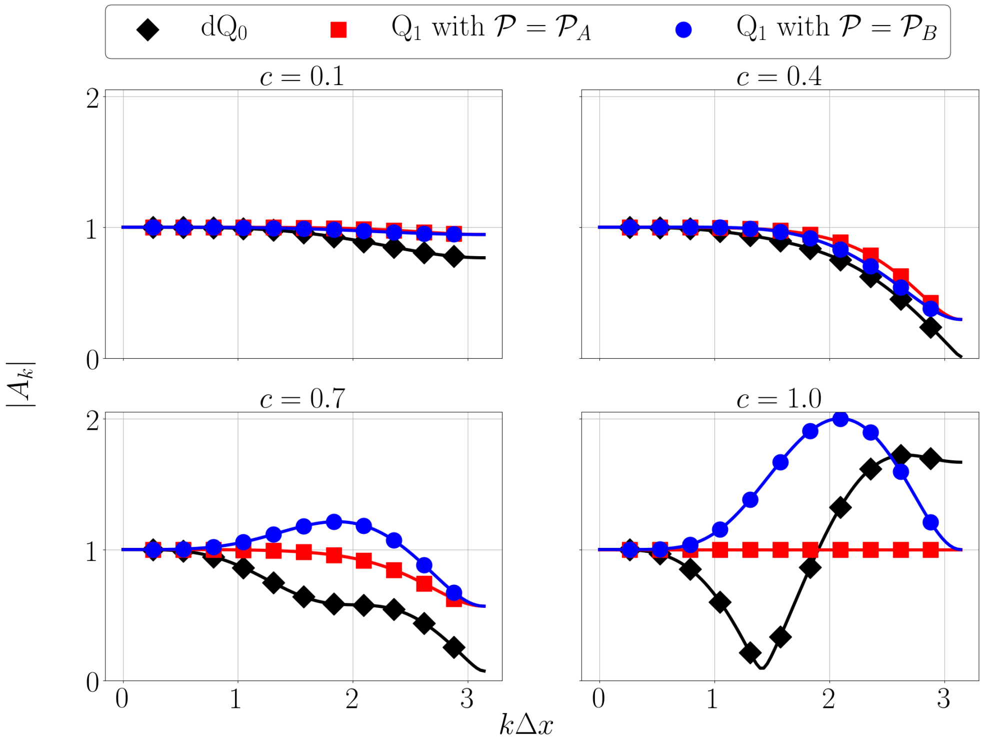


Figure 4.11: The results from testing the validity of the expressions (4.38), (4.40) and (4.42) for the amplification factors in the 1D advection cases presented in Section 4.3.2. The markers denote measurements of the amplification factor by advecting sine and cosine wave profiles, whilst the lines plot the expressions derived in Section 4.3.2. All plots show agreement between the expressions and the measured amplification factors. These results also agree with the critical Courant numbers found in Table 4.2, including for $c = 0.4$ in which the values are all below unity for the Q_1 case with $\mathcal{P}_A$, but with some values above unity for the $\mathcal{P}_B$ case.

attempt to solve is given by

$$\frac{\partial q}{\partial t} = \boldsymbol{\nabla} \cdot (K \boldsymbol{\nabla} \mathbf{q}), \tag{4.54}$$

where K is the diffusivity and $\mathbf{q}$ is the diffused quantity, and may be scalar or vector. When $\mathbf{q}$ is the potential temperature θ_d, we take $\widehat{\mathbf{n}} \cdot \boldsymbol{\nabla}\theta_d$ at the boundary $\partial\Omega$, while with $\boldsymbol{v}$ the boundary condition is $\boldsymbol{v} \cdot \widehat{\mathbf{n}} = 0$ on $\partial\Omega$. In Chapter 3, we described the interior penalty method of [69] to discretise this equation, for use with the $k = 1$ spaces. The formulation in equation (3.25) naturally includes the boundary condition for θ_d, but it must be dealt with explicitly in the solving procedure for $\boldsymbol{v}$. Just as with the DG upwinding scheme for discretising the transport equation, the interior penalty method is not second-order accurate when used with the $k = 0$ spaces. However, the concept of recovering from the $k = 0$ spaces to higher order spaces can be used again to create a second-order diffusion scheme.

Repeating the language of Section 4.2.1, the scheme can be written as the following sequence of operations:

$$q^{n+1} = \mathcal{PDI}(\mathcal{R} - \widehat{\mathcal{P}}\mathcal{R} + 1)q^n. \tag{4.55}$$

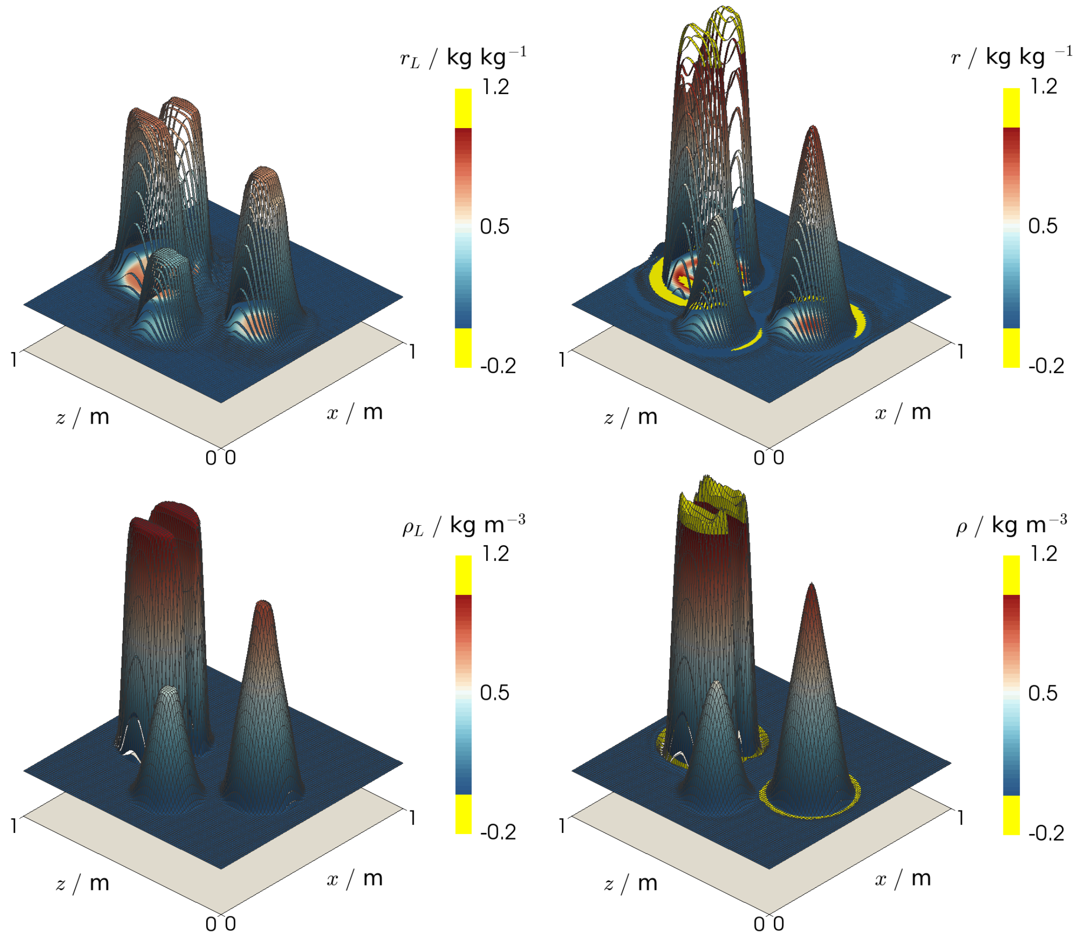


Figure 4.12: The resulting fields from one revolution of the solid-body rotation case of [77]. (Top left) A hypothetical moisture field in the $dQ_0 \otimes Q_1$ space, advected using the limited 'recovered' space scheme, compared with (top right) the same field advected using the non-limited scheme. Although overshoots and undershoots are prevented, conservation of mass is compromised. (Bottom left) A density field in the dQ_1 space, using the same advection operator $\mathcal{A}$ as in the 'recovered' scheme and limited by the vertex-based limiter of [75], with (bottom right) the same solution but without a limiter applied. Overshoots and undershoots are highlighted in yellow, and these are absent from the limited cases, showing the limiter's effectiveness.

The operators $\mathcal{R}$, $\mathcal{I}$, $\mathcal{P}$ and $\widehat{\mathcal{P}}$ are equivalent to those described in Section 4.2.1, and in our model take on (almost) the same form as those described for our transport schemes. The exception is that the projection operator $\mathcal{P}$ must also take into account the boundary condition on $\boldsymbol{v}$ so as not to return velocities that cause outflow or inflow. The spaces used in our model are also equivalent to the spaces laid out in Section 4.2.2 and Table 4.1.

4.6 Test Cases from Compressible Euler Model

To demonstrate the recovered advection scheme, we present results from some common test cases solving the dry compressible Euler equations (3.1). We use the $k = 0$ configuration of Gusto, described thoroughly in Section 3.2, with the recovered scheme used for the advection step.

4.6.1 Rising Thermal

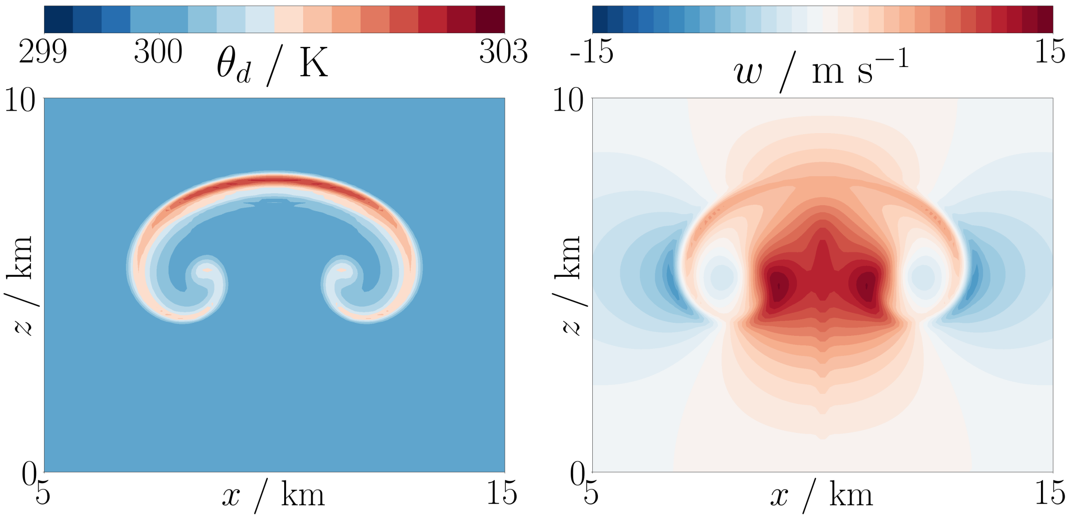


Figure 4.13: Fields at $t = 1000$ s from a run at resolution $\Delta x = 100$ m of the dry thermal case from [78], representing a rising thermal. This is a standard benchmark test in the development of dry dynamical cores. We used the lowest-order family of spaces with the 'recovered' space scheme as the advection method. (Left) the θ_d field, perturbed from a background constant state of 300 K. The field's contour spacing is 0.25 K, and the 300 K contour has been removed for clarity. (Right) The vertical velocity field, contoured every 1 m s^{-1}. These results compare well with those reported in [78].

Here we show some results in the context of a vertical slice model. The example test case that we use is the dry thermal test of [78]. The initial state is $\bar{\theta} = 300$ K and zero velocity everywhere, while ρ_d is determined via solving for hydrostatic balance using the procedure of [62] that was described in Section 3.2.8. The following perturbation to θ_d was then applied:

$$\theta' = \begin{cases} 2\cos^2\left(\pi r/(2r_c)\right) \text{ K}, & r < r_c, \\ 0, & r \geqslant r_c, \end{cases} \tag{4.56}$$

so that $\theta_d = \bar{\theta} + \theta'$, with

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}, \tag{4.57}$$

where $x_c = 10$ km, $z_c = r_c = 2$ km. In the model used in [78], the Exner pressure is a prognostic variable, rather than the density ρ_d. To ensure that our initial pressure is unchanged by the perturbation, we found the initial density state by solving for ρ_{trial}:

$$\int_\Omega \phi \rho_{\text{trial}} \, dx = \int_\Omega \phi \bar{\rho}\bar{\theta}/\theta_d \, dx, \quad \forall \phi \in V_\rho. \tag{4.58}$$

where V_ρ is the function space that ρ_d lives in, and $\bar{\theta}$ and $\bar{\rho}$ are the hydrostatically balanced background states. The domain used had a width of 20 km and a height of 10 km. All boundaries had rigid wall boundary conditions ($\boldsymbol{v} \cdot \widehat{\boldsymbol{n}} = 0$). The Coriolis force is neglected in this simulation and no artificial diffusion is added. The perturbed potential temperature field at the final time $t = 1000$ s is shown in Figure 4.13 for a simulation with grid spacing $\Delta x = 100$ m and time steps of $\Delta t = 1$ s. There is good agreement between the fields shown in Figure 4.13 and those presented in [78].

4.6.2 Baroclinic Wave

Now we show the use of the recovered space transport scheme in our model in three-dimensions with the baroclinic wave test presented by [79]. This test occurs in a channel of length $L = 40000$ km in the x-direction (in which the domain is periodic), width $W = 6000$ km in the y-direction and height $H = 30$ km in the z-direction. The walls at the y and z boundaries of the domain are rigid, with no flow through them. Artificial diffusion is again neglected for both the momentum and thermodynamic equations.

In [79], the analytical initial conditions were provided, describing an atmosphere in thermal wind balance. These are used to define the temperature T, the zonal wind u and the geopotential $\Phi = gz$ in terms of the spatial coordinates (x, y, η), where $\eta = p/p_R$ is a pressure coordinate, with $p_R = 10^5$ Pa. Taking the Coriolis force to be $\boldsymbol{f} = f_0\widehat{\boldsymbol{k}}$, these expressions are

$$u = -u_0 \sin^2\left(\frac{\pi y}{W}\right) \ln\eta \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \tag{4.59a}$$

$$\begin{aligned}\Phi = &\frac{T_0 g}{\Gamma}\left(1 - \eta^{\frac{R_d\Gamma}{g}}\right) \\ &+ \frac{f_0 u_0}{2}\left[y - \frac{W}{2}\left(1 - \frac{1}{\pi}\sin\left(\frac{2\pi y}{W}\right)\right)\right] \ln\eta \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right],\end{aligned} \tag{4.59b}$$

$$\begin{aligned}T = &T_0\eta^{\frac{R_d\Gamma}{g}} \\ &+ \frac{f_0 u_0}{2R_d}\left[y - \frac{W}{2}\left(1 - \frac{1}{\pi}\sin\left(\frac{2\pi y}{W}\right)\right)\right]\left[\frac{2}{b^2}(\ln\eta)^2 - 1\right] \exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right].\end{aligned} \tag{4.59c}$$

The constants take the values $\Gamma = 0.005$ K m^{-1}, $f_0 = 2.00 \times 10^{-6}$ s^{-1}, $a = 6.37 \times 10^6$ m, $T_0 = 288$ K, $u_0 = 35$ m s^{-1} and $b = 2$. The initial meridional and vertical winds are zero.

To use these expressions to find the fields as functions of z rather than η, we use the Newton iteration procedure suggested in [79] to find the initial η field. Using the requirement that $\Phi = gz$ and taking $\eta \in V_\theta$, the $(n+1)$-th estimate of η is given by

$$\eta^{(n+1)} = \eta^{(n)} - \frac{\Phi\left(\eta^{(n)}\right) - gz}{T\left(\eta^{(n)}\right) - R_d/\eta^{(n)}}, \tag{4.60}$$

where $T\left(\eta^{(n)}\right)$ and $\Phi\left(\eta^{(n)}\right)$ are the expressions from (4.59). Once this has converged, η can be used with (4.59) to find T and u. The prognostic variable θ_d is obtained by combining T with $p = p_R\eta$, which is then used with the hydrostatic balance procedure of Section 3.2.8 to acquire the initial ρ_d.

The perturbation is added to the x-component of the background velocity, which we denote with $\bar{u}$, so that $u = \bar{u} + u'$. With $x_c = 2 \times 10^6$ m, $y = 2.5 \times 10^6$ m, $L_p = 6 \times 10^5$ and $u_p = 1$ m s^{-1}, we use

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \tag{4.61}$$

to get

$$u' = u_p \exp\left[-\left(\frac{r}{L_p}\right)^2\right]. \tag{4.62}$$

Figure 4.14 shows fields at $t = 12$ days from a simulation using the $k = 0$ configuration of Gusto with $\Delta x = \Delta y = 200$ km, $\Delta z = 1$ km and $\Delta t = 300$ s. Although the comparable figures from [79] were performed by a model at higher resolution, our results show a good resemblance, suggesting our model is describing the baroclinic wave well.

4.6.3 Non-linear Density Current

As a demonstration of the recovered diffusion scheme, we present results using Gusto in the $k = 0$ configuration of the non-linear density current test of [80]. This is a vertical slice test, with a domain of length 25.6 km and height 6.4 km. The side walls are now rigid boundaries, so only velocities with $\boldsymbol{v} \cdot \widehat{\boldsymbol{n}} = 0$ are accepted solutions. The initial condition is found by setting $\theta_d = 300$ K everywhere, and using this to find the hydrostatically balanced ρ_d using the procedure outlined in Section 3.2.8, subject to $\Pi = 1$ on the bottom boundary. A perturbation is then applied, so that the initial θ_d at the beginning of the model is

$$\theta_d = \begin{cases} 300 \text{ K}, & r > 1, \\ \left[300 - \frac{15}{2\Pi}(1 + \cos(\pi r))\right] \text{ K}, & r \leqslant 1, \end{cases} \tag{4.63}$$

where r is given by

$$r^2 = \left(\frac{x - x_c}{x_r}\right)^2 + \left(\frac{z - z_c}{z_r}\right)^2, \tag{4.64}$$

where $x_c = 0$ km, $z_c = 4$ km, $x_r = 4$ km and $z_r = 2$ km. The initial velocity is zero everywhere and again the Coriolis force is neglected. The diffusion terms were added to the equations for $\boldsymbol{v}$ and θ_d, using $K = 75$ m^2 s^{-1} for both equations. Using the interior penalty scheme described in Section 3.2.7, we took $\mu = 10/\Delta x$ for both $\boldsymbol{v}$ and θ_d equations.

Results shown in Figure 4.15 are taken at $t = 900$ s for a simulation with $\Delta x = \Delta z = 100$ and $\Delta t = 1$ s. These results compare favourably with other documented cases, such as those described in [10], with some separation between the vortices being resolved.

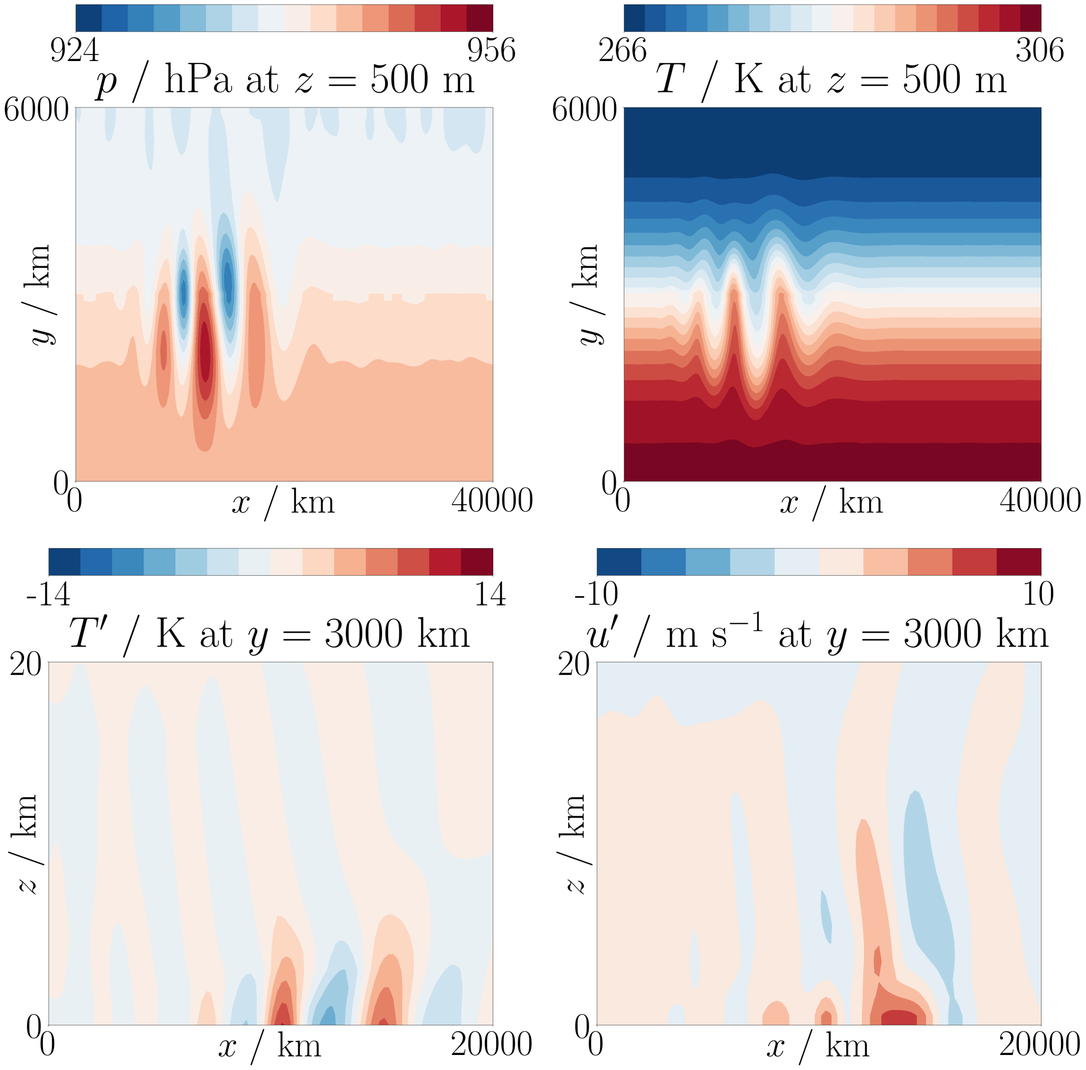


Figure 4.14: Fields from the dry baroclinic wave simulation at $t = 12$ days, computed with $\Delta x = \Delta y = 200$ km, $\Delta z = 1$ km and $\Delta t = 300$ from the $k = 0$ configuration of Gusto. (Top left) the pressure field at $z = 500$ m contoured every 2 hPa, (top right) the temperature field at $z = 500$ m contoured every 2 K, (bottom left) the perturbed meridional wind u' at $y = 3000$ km, with contour spacing 5 m s^{-1} and (bottom right) the perturbed temperature field at $y = 3000$ km with contours every 2 K. Since both of the fields shown on the $y = 3000$ km plane are discontinuous there, the values shown are computed from the lower side of the plane. This test case demonstrates the use of the lowest-order elements in three-dimensions and in describing the Coriolis force. With cells with a high aspect (large horizontal length but small vertical depth), this test mimics the equations solved in a numerical weather prediction model, save on a Cartesian domain.

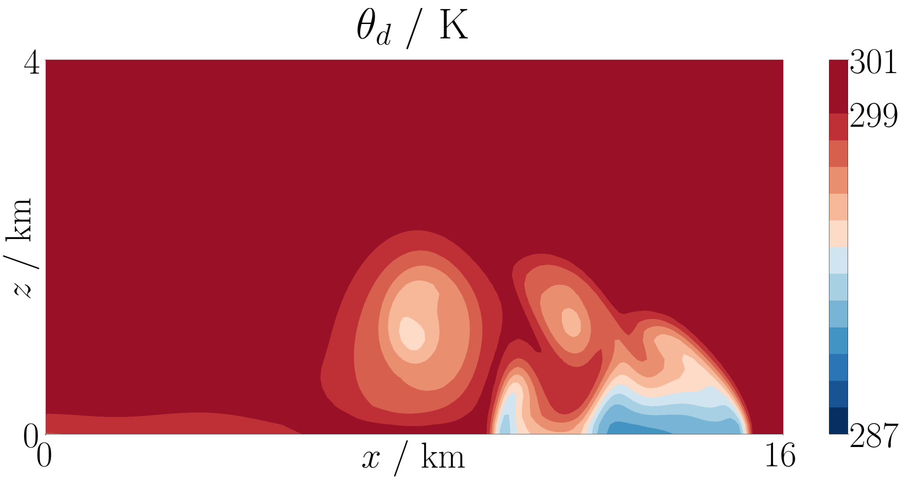


Figure 4.15: The θ_d field at $t = 900$ s from the density current test case of [80]. This simulation was performed with $\Delta x = \Delta z = 100$ m. The contour spacing is 1 K, and the 300 K contour has been removed for clarity. This density current test case is another standard test in the development of dynamical cores, and is used to illustrate our implementation of the artifical diffusion terms for the lowest-order spaces.

4.7 Summary and Outlook

In this chapter we have presented a new 'recovered' advection scheme in the context of atmospheric simulations. This scheme is a form of the embedded DG advection described in [67], but in which higher-degree spaces are recovered via averaging operators, as described in [71]. We presented this scheme in the context of the compatible finite element set-up laid out in [8], and in particular so that it can be used with the zeroth-degree set of spaces. With these spaces, the scheme has second-order numerical accuracy. We have also presented a bounded version of this scheme, which can be used for moisture variables to preserve monotonicity or to prevent negative values, and also applied the same principle to a diffusion scheme. Stability properties of the scheme have also been provided and demonstrated with numerical results, and examples of the transport scheme within Gusto have been shown through three test cases.

The main strength of the scheme is that it facilitates the use of the $k = 0$ lowest-order spaces in Gusto, which offer advantages for coupling the model to the physics parametrisations which often involve decomposing the fields into mean and fluctuating parts. It also gives a means of direct comparison to other models such as [10] that use the lowest-order spaces. There is no degradation in the stability relative to the DG upwinding scheme on linear discontinuous spaces, and as we will see in the following chapter, in some test cases the $k = 0$ spaces compare more favourably to the literature results than the $k = 1$ spaces.

However, as the scheme involves various recovery and projection steps and because the advection itself occurs at higher resolution than the rest of the model, the scheme can be computationally expensive. Indeed in three-dimensions dQ_1 has eight times more DOFs than dQ_0. Although this may result in increased accuracy relative to other second-order transport schemes, semi-Lagrangian or second-order finite volume schemes can be much cheaper to use. The lowest-order spaces also may not necessarily provide increases in efficiency in the linear-solve part of the model compared with the hybridised approach discussed in Section 3.2.6. An alternative approach could be to perform all of the dynamics stage of the model using the $k = 1$ spaces before projecting the fields into the $k = 0$ spaces to compute the physics parametrisations. The recovery operators can then be used to restore the field to the $k = 1$ spaces. This approach might be supported by [81], which advocates calculating the parametrisations on a lower resolution grid. Indeed, one of the main consequences of the work presented in this chapter is the demonstration that performing different stages of the model with different function spaces may be a viable option, whilst using the recovery operators allows this to be done with higher accuracy.

For the use of this transport scheme in the future, within Gusto or elsewhere, several augmentations are necessary. Firstly, the scheme needs to be extended for the transport of vector fields over curved manifolds, which will require careful averaging of vectors that lie in different tangent spaces. To provide for non-regular meshes, the scheme should use weighted averaging, with the weights provided by the distances to the respective DOFs corresponding the values used in the recovery. The procedure for additional recovery at the boundaries is currently only supported on quadrilateral meshes where the topography is shallow, whilst more sophisticated limiting techniques could be explored that are less liable to degrade the solution.

5 A Compatible Finite Element Method for a Moist Atmosphere

In this chapter, we present a discretisation of the moist compressible Euler equations within Gusto. The broader goal is to investigate the consequences upon physics-dynamics coupling of using a dynamical core with a compatible finite element discretisation. A natural first parametrisation to include concerns moisture, which is tightly coupled with the dry dynamics and obeys well-known physical laws. In this chapter, building upon the discretisation presented in Chapter 3, we describe the new features added to Gusto whilst implementing the discretisation. Results from test cases then showcase the discretisation. This chapter thus represents one of the first implementations of physics within the compatible finite element framework.

The main results of this chapter are:

- presentation of the details of the discretisation of the moist compressible Euler equations within Gusto, representing a major milestone in Gusto's development;
- routines for establishing hydrostatic balance in moist conditions, which is important for establishing the initial conditions for many test cases;
- discussion of strategies for combining functions from different finite element spaces for physics parametrisations;
- presentation of a limiter for moisture variables in the $k = 1$ configuration, which prevents the formation of undershoots and overshoots;
- presentation of a new moist variant of the dry gravity wave test case of [82], which can be used to show the convergence properties of the model;
- presentation of results of a new 3D saturated atmosphere test, which extends the benchmark test of [78] and contributes to addressing the gap in common test cases between simple dry tests and those using a full suite of physics parametrisations;
- comparison of the $k = 0$ and $k = 1$ configurations of Gusto through some moisture test cases, which shows some significant differences between using each set of elements.

This chapter is adapted from our paper [14], where many of these results are also presented.

5.1 Governing Equations

We solve the compressible Euler equations featuring three species of moisture: water vapour, cloud water and rain. The ice phase is neglected. Motivated by the UK Met Office's most recent dynamical core, ENDGame, [66], our prognostic variables still include wind velocity $\boldsymbol{v}$ and the density of dry air ρ_d, but now also the dry virtual potential temperature θ_{vd} and the mixing ratios r_v, r_c and r_r. Here the subscripts respectively denote water vapour, cloud water and rain, whilst the mixing ratio r_i is the ratio of the density by volume of the i-th substance to that of dry air, i.e. $r_i := \rho_i/\rho_d$. The total mixing ratio of water species is $r_t := r_v + r_c + r_r$. The dry virtual potential temperature θ_{vd} is defined for r_v, temperature T and air pressure p by

$$\theta_{vd} := T\left(\frac{p_R}{p}\right)^{\frac{R_d}{c_{pd}}}(1 + r_v/\epsilon) = \theta_d(1 + r_v/\epsilon), \tag{5.1}$$

where $\epsilon := R_d/R_v$ is the ratio of specific gas constants of dry air and water vapour. Thus in the absence of water vapour, θ_{vd} reduces to θ_d. The choice of θ_{vd} is motivated in [66], which notes that it is the more natural choice of variable to complement ρ_d, and claims that it may be smoother than the dry potential temperature θ_d.

The full equation set that we use is

$$\frac{D\boldsymbol{v}}{Dt} + \mathbf{f} \times \boldsymbol{v} + \frac{c_{pd}\theta_{vd}}{1 + r_t}\boldsymbol{\nabla}\Pi + \boldsymbol{\nabla}\Phi = \boldsymbol{\nabla} \cdot (K\boldsymbol{\nabla}\boldsymbol{v}), \tag{5.2a}$$

$$\begin{aligned}\frac{D\theta_{vd}}{Dt} + \theta_{vd}\left(\frac{R_m}{c_{vml}} - \frac{R_d c_{pml}}{c_{pd}c_{vml}}\right)\boldsymbol{\nabla} \cdot \boldsymbol{v} &= \boldsymbol{\nabla} \cdot (K\boldsymbol{\nabla}\theta_{vd}) \\ -\theta_{vd}\left[\frac{c_{vd}L_v(T)}{c_{vml}c_{pd}T} - \frac{R_v}{c_{vml}}\left(1 - \frac{R_d c_{vml}}{R_m c_{pd}}\right) - \frac{R_v}{R_m}\right]\frac{Dr_v}{Dt},\end{aligned} \tag{5.2b}$$

$$\frac{D\rho_d}{Dt} + \rho_d\boldsymbol{\nabla} \cdot \boldsymbol{v} = 0, \tag{5.2c}$$

$$\frac{Dr_v}{Dt} = -\dot{r}^c_{\text{cond}} + \dot{r}^r_{\text{evap}}, \tag{5.2d}$$

$$\frac{Dr_c}{Dt} = \dot{r}^c_{\text{cond}} - \dot{r}_{\text{accr}} - \dot{r}_{\text{accu}}, \tag{5.2e}$$

$$\frac{Dr_r}{Dt} = \dot{r}_{\text{accr}} - \dot{r}_{\text{accu}} - \dot{r}^r_{\text{evap}} - S. \tag{5.2f}$$

The notation is as before in equations (3.1), though the forcing terms on the right hand side have now been omitted. The advection terms have been absorbed in the advective derivative, which is given by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + \boldsymbol{v} \cdot \boldsymbol{\nabla}. \tag{5.3}$$

For the specific heat capacities c_{pd}, c_{vd}, c_{vml} and c_{pml}, the specific gas constant R_m and also the latent heat of vaporization $L_v(T)$, we follow closely the values used in [78], which are also listed in the appendix. The diffusion terms are retained in this equation, although they will not be used in this chapter.

The equation of state is now given by

$$\Pi = \left(\frac{p}{p_R}\right)^{\kappa} \equiv \left(\frac{\rho_d R_d \theta_{vd}}{p_R}\right)^{\frac{\kappa}{1-\kappa}}, \tag{5.4}$$

with $\kappa := R_d/c_{pd}$, as before.

The terms on the left hand sides of (5.2) represent the dynamics, whilst the right hand sides are considered to be the physics. The processes $\dot{r}^c_{\text{cond}}$, $\dot{r}_{\text{accr}}$, $\dot{r}^r_{\text{evap}}$, $\dot{r}_{\text{accu}}$ and S are the microphysics parametrisations and are described in Section 5.4.

The final thing to note is the extra term proportional to $\nabla \cdot \boldsymbol{v}$ appearing on the left hand side of (5.2b). This term is neglected in many models but mentioned in [83] and [78] to be important in fully capturing convection, particularly in a saturated atmosphere. In our model it appears in the forcing step of the dynamical core.

5.2 Dynamics Discretisation

In this section we will illustrate the differences to the dynamical core part of the model following the addition of moisture. The prognostic variables are now expanded, so that $\chi = (\boldsymbol{v}, \rho_d, \theta_{vd}, r_v, r_c, r_r)$. The dry virtual potential temperature θ_{vd} essentially assumes the same role as the potential temperature θ_d, and will lie in the same space V_θ. The moisture variables r_v, r_c and r_r are all co-located with θ_{vd} in V_θ, as dynamically they can be considered as an adjustment to θ_{vd} in the momentum equation, and so as to facilitate latent heat transfer as water condenses or evaporates. The physics parametrisations, involving the microphysics processes, are added at the end of the time step. These are described in Section 5.4. To represent finding the tendencies from these parametrisations we use the operator $\mathcal{P}(\chi)$. The pseudocode from Section 3.2.1 now becomes

```
1. FORCING: $\chi^* = \chi^n + (1-\alpha)\Delta t \mathcal{F}(\chi^n)$
2. SET: $\chi_p^{n+1} = \chi^n$
3. OUTER:
    (a) UPDATE: $\bar{\boldsymbol{u}} = \alpha \boldsymbol{v}_p^{n+1} + (1-\alpha)\boldsymbol{v}^n$
    (b) ADVECT: $\chi_p = \mathcal{A}_{\bar{u}}(\chi^*)$
    (c) INNER:
        i. FIND RESIDUAL: $\Delta\chi = \chi_p + \alpha\Delta t \mathcal{F}(\chi_p^{n+1}) - \chi_p^{n+1}$
        ii. SOLVE: $\mathcal{S}(\chi') = \Delta\chi$ for $\chi'$
```

iii. `INCREMENT:` $\chi_p^{n+1} = \chi_p^{n+1} + \chi'$

4. `DIFFUSION:` $\chi_p^{n+1} = \mathcal{D}(\chi_p^{n+1})$

5. `PHYSICS:` $\chi_p^{n+1} = \mathcal{P}(\chi_p^{n+1})$

6. `ADVANCE TIME STEP:` $\chi^n = \chi_p^{n+1}$

Changes to the forcing and linear solve steps are described in the rest of the section.

5.2.1 Forcing

The 'forcing' step, described in Section 3.2.4, must now take into account the presence of moisture. This simply involves modifying equation (3.10). For both the $k = 0$ and $k = 1$ cases, $\mathcal{F}(\boldsymbol{v})$ is the $\boldsymbol{v}_{\text{trial}}$ that solves, for all $\boldsymbol{\psi} \in V_v$,

$$\begin{aligned}\int_\Omega \boldsymbol{\psi} \cdot \boldsymbol{v}_{\text{trial}} \, dx = \int_\Omega \left[c_{pd} \boldsymbol{\nabla} \cdot \left(\frac{\theta_{vd} \boldsymbol{\psi}}{1 + r_t} \right) \Pi - \boldsymbol{\psi} \cdot (\boldsymbol{f} \times \boldsymbol{v}) \right] \, dx \\ - \int_\Omega g(\boldsymbol{\psi} \cdot \widehat{\boldsymbol{k}}) \, dx - \int_\Gamma c_{pd} \left[\!\!\left[\frac{\theta_{vd} \boldsymbol{\psi}}{1 + r_t} \right]\!\!\right]_{\boldsymbol{n}} \langle \Pi \rangle \, dS.\end{aligned} \tag{5.5}$$

The more significant difference between the dry and moist cases is that equation (5.2b) contains non-advective terms, which are included in the 'forcing' step. This means finding $\mathcal{F}(\theta_{vd})$, which is the solution θ_{trial}, for all $\gamma \in V_\theta$ to

$$\int_\Omega \gamma \theta_{\text{trial}} \, dx = - \int_\Omega \gamma \theta_{vd} \left(\frac{R_m}{c_{vml}} - \frac{R_d c_{pml}}{c_{pd} c_{vml}} \right) (\boldsymbol{\nabla} \cdot \boldsymbol{v}) \, dx. \tag{5.6}$$

This is the same for both the $k = 0$ and $k = 1$ configurations.

5.2.2 Linear Solve

With the introduction of moisture, the linear solve step for the implicit stage of the model slightly changes from that described in Section 3.2.6. The linearised equations, that are solved given a residual, become

$$\boldsymbol{v}' + \frac{\alpha \Delta t c_{pd}}{1 + r_t} \left(\theta'_{vd} \boldsymbol{\nabla} \overline{\Pi} + \overline{\theta}_{vd} \boldsymbol{\nabla} \Pi' \right) = \Delta \boldsymbol{v}, \tag{5.7a}$$

$$\rho'_d + \alpha \Delta t \boldsymbol{\nabla} \cdot \left(\overline{\rho}_d \boldsymbol{v}' \right) = \Delta \rho_d, \tag{5.7b}$$

$$\theta'_{vd} + \alpha \Delta t \left(\widehat{\boldsymbol{k}} \cdot \boldsymbol{\nabla} \overline{\theta}_{vd} \right) \left(\widehat{\boldsymbol{k}} \cdot \boldsymbol{v}' \right) = \Delta \theta_{vd}. \tag{5.7c}$$

Two new approximations have been made in (5.7): perturbations in moisture fields have been neglected, as is the non-advective forcing term featuring in (5.6). Now θ'_{vd} takes the role of θ'_d, being eliminated by the definition of Θ':

$$\Theta' = \Delta\theta_{vd} - \alpha\Delta t\left(\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\bar{\theta}_{vd}\right)\left(\hat{\boldsymbol{k}}\cdot\boldsymbol{v}'_{\text{trial}}\right). \tag{5.8}$$

It also appears in the approximation of Π':

$$\Pi' = \frac{\kappa}{1-\kappa}\bar{\Pi}\left(\frac{\Theta'}{\bar{\theta}_{vd}} + \frac{\rho'_{\text{trial}}}{\bar{\rho}_d}\right). \tag{5.9}$$

The moisture species also appear in the mixed system, which is then solved to find $\left(\boldsymbol{v}'_{\text{trial}}, \rho'_{\text{trial}}, \ell'_{\text{trial}}\right) \in \left(\hat{V}_{\boldsymbol{v}}, V_\rho, V_{\text{trace}}\right)$, for all $(\boldsymbol{\psi}, \phi, \lambda) \in \left(\hat{V}_{\boldsymbol{v}}, V_\rho, V_{\text{trace}}\right)$. Then, equation (3.23) becomes

$$\begin{aligned}
&\int_\Omega \boldsymbol{\psi}\cdot\left(\boldsymbol{v}'_{\text{trial}} - \Delta\boldsymbol{v}\right)\,dx \\
&-\alpha\Delta t c_{pd}\left(\int_\Omega \boldsymbol{\nabla}\cdot\left[\frac{\Theta' w}{1+r_t}\hat{\boldsymbol{k}}\right]\bar{\Pi}\,dx - \int_\Gamma \left[\!\!\left[\frac{\Theta' w}{1+r_t}\hat{\boldsymbol{k}}\right]\!\!\right]_{\boldsymbol{n}}\langle\bar{\Pi}\rangle\,dS\right) \\
&+\alpha\Delta t c_{pd}\left(\int_{\partial\Omega}\frac{\Theta' w}{1+r_t}\hat{\boldsymbol{k}}\cdot\hat{\boldsymbol{n}}\langle\bar{\Pi}\rangle\,ds - \int_\Omega \boldsymbol{\nabla}\cdot\left[\frac{\bar{\theta}_{vd}}{1+r_t}\boldsymbol{\psi}\right]\Pi'\,dx\right) \\
&+\alpha\Delta t c_{pd}\left(\int_\Gamma \left[\!\!\left[\frac{\bar{\theta}_{vd}}{1+r_t}\boldsymbol{\psi}\right]\!\!\right]_{\boldsymbol{n}}\ell'_{\text{trial}}\,dS + \int_{\partial\Omega}\frac{\bar{\theta}_{vd}}{1+r_t}\left(\boldsymbol{\psi}\cdot\hat{\boldsymbol{n}}\right)\ell'_{\text{trial}}\,ds\right) \\
&+\int_\Omega \phi\left(\rho'_{\text{trial}} - \Delta\rho_d\right)\,dx - \alpha\Delta t\int_\Omega\left(\boldsymbol{\nabla}\phi\cdot\boldsymbol{v}'_{\text{trial}}\right)\bar{\rho}_d\,dx \\
&+\alpha\Delta t\left(\int_\Gamma \left[\!\left[\phi\boldsymbol{v}'_{\text{trial}}\right]\!\right]_{\boldsymbol{n}}\langle\rho_d\rangle\,dS + \int_{\partial\Omega}\phi\boldsymbol{v}'_{\text{trial}}\cdot\hat{\boldsymbol{n}}\langle\rho_d\rangle\,ds\right) \\
&+\int_\Gamma \lambda\left[\!\left[\boldsymbol{v}'_{\text{trial}}\right]\!\right]_{\boldsymbol{n}}\,dS + \int_{\partial\Omega}\lambda\left(\boldsymbol{v}'_{\text{trial}}\cdot\hat{\boldsymbol{n}}\right)\,ds = 0.
\end{aligned} \tag{5.10}$$

The dry virtual potential temperature θ_{vd} is restored by finding θ'_{trial} that solves, for all $\gamma \in V_\theta$,

$$\int_\Omega \gamma\left[\theta'_{\text{trial}} - \Delta\theta_{vd} + \alpha\Delta t\left(\hat{\boldsymbol{k}}\cdot\boldsymbol{v}\right)\left(\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\bar{\theta}_{vd}\right]\right) = 0. \tag{5.11}$$

5.2.3 Limiting

As discussed in Section 4.2.4, the advection schemes that we use do not preserve the monotonicity property of the continuous transport equation. This can result in unphysical negative values when advecting moisture variables. The way we prevent this is through the application of a slope-limiter, and the limiting strategies that we use are based upon the vertex-based limiter of [75] that we described in Section 4.2.4.

In Chapter 4, we described that, in the $k = 0$ configuration, the approach to limiting the advection of moisture variables was to apply the vertex-based limiter of [75]. This is applied to each step of the transport in the dQ_1 space. This is augmented by a different projection operator (which we called $\mathcal{P}_B$) for the projection back to the $k = 0$ spaces once the advection was complete.

With the $k = 1$ spaces, the vertex-based limiter of [75] is again used. However moisture fields in V_θ are quadratic in the vertical and the values of the field at vertical midpoint of the cell are unchanged by the vertex-based limiter. We bound these values by restricting them to the average of the values at the adjacent vertices if the midpoint values fall outside of the range spanned by the vertex values. In other words, if $r_{i+1/2}$ is the value of the field at a degree of freedom halfway up the i-th cell, then the new value is

$$r_{i+1/2} = \begin{cases} r_{i+1/2} & \text{if } \min\left(r_i, r_{i+1}\right) \leqslant r_{i+1/2} \leqslant \max\left(r_i, r_{i+1}\right), \\ \frac{1}{2}\left(r_i + r_{i+1}\right), & \text{otherwise.} \end{cases} \tag{5.12}$$

The advection of fields in V_θ for the $k = 1$ configuration uses the embedded advection scheme of [67]. As described in Section 3.2.5, the final stage of this scheme is to convert it from the embedded space back into the original space via a Galerkin projection. This can result in overshoots and undershoots, so when the advection is to be limited, this step is performed using a recovery operator.

To illustrate the effectiveness of this limiter, we performed the rotation of the slotted-cylinder, hump, cone of [77] as in Section 4.4.3. Figure 5.1 shows the fields in the V_θ space for the $k = 1$ configuration, rotated once around the domain by a velocity defined by the stream function

$$\psi = \pi(z(z-1) + x(x-1)). \tag{5.13}$$

The domain used was the unit square, with $\Delta x = \Delta z = 0.01$, whilst the time step was $\Delta t = 10^{-4}$. The final fields were recorded at $t = 1$. This limiter shows good performance, avoiding the degradation of mass observed with the limiter for the $k = 0$ spaces seen in Section 4.4.3. If improvements were required, a more sophisticated limiter for transport in the $k = 1$ V_θ space is described in [67].

5.3 Physics-Dynamics Coupling

5.3.1 Combining Fields From Different Function Spaces

Many diagnostic fields, such as temperature T or the saturation mixing ratio of water vapour r_{sat} must be determined from prognostic fields in different function spaces, usually ρ_d, θ_{vd} and r_v. Such diagnostic fields are important in the equations used for the physics parametrisations and can also be important in setting the initial state of the model.

As ρ_d and θ_{vd} lie in different function spaces, when determining some diagnostic variable q there are a number of different choices that can be taken. With our emphasis on

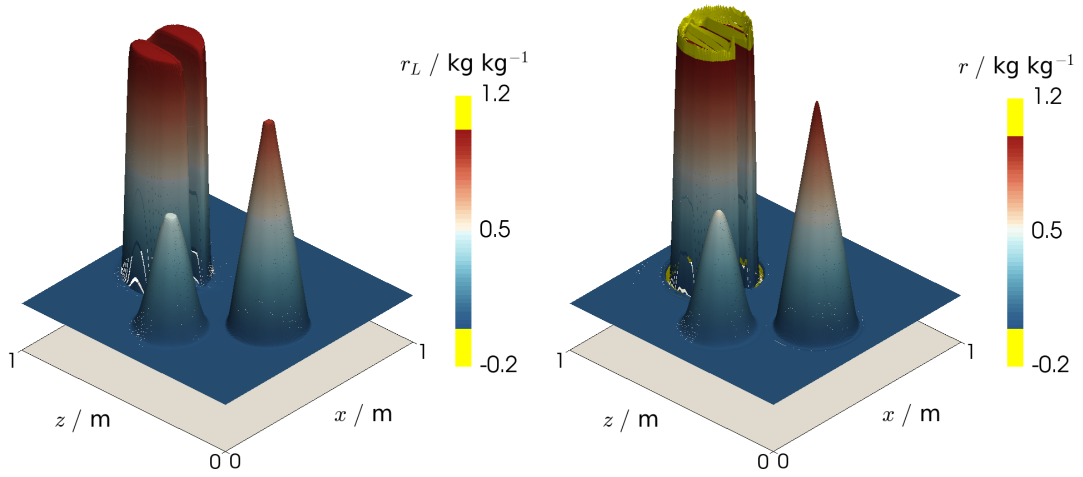


Figure 5.1: The final fields, following one rotation around the domain, of functions in the two-dimensional $dQ_1 \otimes Q_2$ space that were initialised with the slotted-cylinder, hump, cone initial condition of [77]. (Left) with the limiter applied, and (right) without the limiter. Overshoots and undershoots are highlighted in yellow, showing that the limiter does indeed prevent overshoots and undershoots from forming. The limiter also does not appear to impact heavily upon the magnitude of the profile.

parametrisations of moist processes, we take only the case of $q \in V_\theta$, so that it is also in the same function space as the moisture variables. We considered three approaches to determining q from the prognostic variables:

1. *Recovering density*: the density field ρ_d is 'recovered' into V_θ using the recovery operator outlined in Section 4.2.3 to give $\widetilde{\rho}_d$. Then q can be calculated algebraically within V_θ. For the $k = 0$ spaces, the recovery operator requires the extra recovery at the boundaries that was described in Section 4.2.3.

2. *Projecting density*: the density field is approximated in V_θ via a Galerkin projection. This gives the $\widetilde{\rho}_d \in V_\theta$ that solves for all $\psi \in V_\theta$

$$\int_\Omega \psi \widetilde{\rho}_d \, dx = \int_\Omega \psi \rho_d \, dx, \tag{5.14}$$

   where Ω is the domain. The projected field $\widetilde{\rho}_d$ is then used to algebraically find q.

3. *Weak evaluation*: q is projected directly from its expression into V_θ. This involves finding $q \in V_\theta$ that solves for all $\psi \in V_\theta$

$$\int_\Omega \psi q \, dx = \int_\Omega \psi q(\theta_{vd}, \rho_d, r_v) \, dx. \tag{5.15}$$

To compare these approaches, we considered smooth analytical forms for T, p and r_v fields in a two-dimensional vertical slice domain, with height and width $L = 10$ km.

$$T = 280 - 5\cos\left(\frac{2\pi z}{L}\right) + 5\left(\frac{x}{L}\right), \tag{5.16a}$$

$$p = 10^5 e^{-z/2000}\left[1 + \frac{1}{100}\cos\left(\frac{2\pi x}{L}\right)\right], \tag{5.16b}$$

$$r_v = \frac{1}{200} + \frac{1}{1000}\cos\left(\frac{2\pi z}{L}\right)\sin\left(\frac{2\pi z}{L}\right). \tag{5.16c}$$

These can then be used to compute analytical expressions for q and also the prognostic variables, via

$$\theta_{vd} = T\left(\frac{p_R}{p}\right)^{\kappa}(1 + r_v/\epsilon), \quad \rho_d = \frac{p}{R_d T(1 + r_v/\epsilon)}. \tag{5.17}$$

These analytical expressions are interpolated into spaces of high polynomial degree, before projecting them to get $q, \theta_{vd} \in V_\theta$ and $\rho_d \in V_\rho$. The three approaches above are then used for both the $k = 0$ and $k = 1$ sets of spaces to estimate q from the discrete θ_{vd} and ρ_d fields. These estimations of q are compared with the q that was determined directly from the analytic expressions. For q we chose the wet-equivalent potential temperature outlined in equation (5.37).

The error between the estimated θ_e fields with the analytic θ_e are measured at different resolutions and plotted in Figure 5.2. In the $k = 1$ setup there is little difference between any of the methods, but in the $k = 0$ case the *recovering density* approach creates solutions which converge better and with smaller errors. The other approaches cannot represent the field at the boundary so accurately and thus are worse. We therefore use this approach to recover the density into V_θ, where it is used to calculate other diagnostic fields.

5.3.2 Temporal Coupling of Physics to Dynamics

One of the most important aspects of physics-dynamics coupling within an atmospheric model is where the physics routines are run in the model relative to the dynamics. Currently in our model, the physics routines are called once per time step, after the dynamics time step has been completed. Another important choice is the order in which the physics routines are run. In our model the processes are split *sequentially*, so that the final state of one physics routine becomes the initial state of the next physics routine. These are performed in the following order:

1. sedimentation of rain;

2. accretion of cloud water and auto-accumulation of rain water;

3. evaporation of rain water;

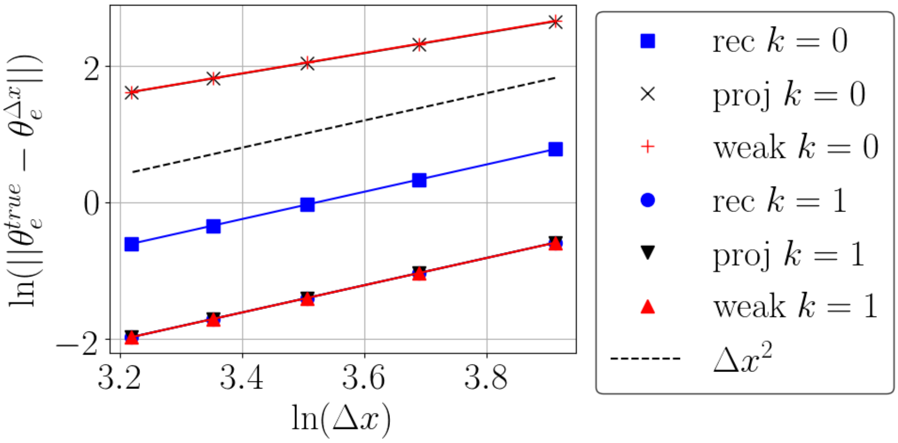


Figure 5.2: Numerical results showing the error and its convergence for three different methods of combining functions from different function spaces to determine a parametrisation. The three methods are: *recovering* ρ_d into V_θ, *projecting* ρ_d into V_θ or calculating θ_e directly through a *weak formulation*. Results are shown measuring the error between an analytic θ_e field and one calculated from ρ_d, θ_{vd} and r_v using these methods. The results are in both the $k = 0$ and $k = 1$ spaces. For the $k = 1$ spaces, there is very little difference, but for the $k = 0$ spaces only the recovery operator can accurately represent the fields at the boundaries of the domain. These results lead us to use recovery to convert ρ_d into V_θ for use in physical parametrisations.

4. evaporation of cloud water/condensation of water vapour.

We choose to do the evaporation/condensation step last, so as to prevent any supersaturation at the end of the time step.

5.4 Physics Parametrisations

In this section we consider the discretisation of the 'physics' terms in equation set (5.2). Each term is evaluated individually in a separate 'physics' routine. The term labelled $\dot{r}^c_{\text{cond}}$ represents the condensation of water vapour into cloud water and the evaporation of cloud water into water vapour. The similar term $\dot{r}^r_{\text{evap}}$ describes the evaporation of rain water into water vapour, whilst $\dot{r}_{\text{accr}}$ and $\dot{r}_{\text{accu}}$ are known as the accretion and auto-accumulation processes which convert cloud water into rain. The sedimentation of rain is given by S. The details of each of these terms are presented in the following section.

5.4.1 Combining Functions from Different Spaces

Many of the parametrisations depend upon the fields T and p which are prognostic variables. We currently attempt to avoid direct evaluation of these fields and write them as functions of our prognostic variables:

$$T = \frac{\theta_{vd}\widetilde{\Pi}}{1 + r_v/\epsilon}, \qquad \widetilde{p} = p_R \widetilde{\Pi}^{\frac{1}{\kappa}}, \tag{5.18}$$

where

$$\widetilde{\Pi} = \left(\frac{\widetilde{\rho}_d R_d \theta_{vd}}{p_R}\right)^{\frac{\kappa}{1-\kappa}}. \tag{5.19}$$

Here we denote fields with the tilde to emphasise that they are considered to lie in the temperature space.* The method used to obtain $\widetilde{\rho}_d$ is described further in Section 5.3.1. As was mentioned in that section, the conversion of ρ_d to $\widetilde{\rho}_d$ could be avoided by writing the parametrisations in weak form (multiplying by a test function before integrating). Following the results of Section 5.3.1, now we do not take this approach.

5.4.2 Condensation/Evaporation

The condensation/evaporation rate between water vapour and cloud water is that used in [84], and [78]:

$$\dot{r}^0_{\text{cond}} := \frac{r_v - r_{\text{sat}}(\widetilde{p}, T)}{\Delta t \left(1 + \dfrac{L_v^2(T) r_{\text{sat}}(\widetilde{p}, T)}{c_{pd} R_v T^2}\right)}, \tag{5.20}$$

where Δt is the time step and $r_{\text{sat}}(\widetilde{p}, T)$ is the saturation value of r_v to be defined below. We use the following restrictions to prevent generation of negative values of r_v or r_c:

$$\dot{r}^c_{\text{cond}} = \begin{cases} \max(\dot{r}^0_{\text{cond}}, -r_c/\Delta t) & \dot{r}^0_{\text{cond}} < 0, \\ \min(\dot{r}^0_{\text{cond}}, r_v/\Delta t), & \dot{r}^0_{\text{cond}} > 0. \end{cases} \tag{5.21}$$

The saturation value of r_v that we use stems from Tetens' empirical formula [85], and is fully described in [86]. The formula given for the saturation water vapour pressure $e_{\text{sat}}(T)$ is:

$$e_{\text{sat}}(T) = C_0^{\text{sat}} \exp\left[\frac{C_1^{\text{sat}}(T - T_R)}{T - C_2^{\text{sat}}}\right], \tag{5.22}$$

where C_0^{sat}, C_1^{sat} and C_2^{sat} are constants whose values are given in the appendix. This can be used to find $r_{\text{sat}}(\widetilde{p}, T)$:

$$r_{\text{sat}}(\widetilde{p}, T) = \frac{\epsilon C_0^{\text{sat}}}{\widetilde{p} \exp\left[-\dfrac{C_1^{\text{sat}}(T - T_R)}{T - C_2^{\text{sat}}}\right] - C_0^{\text{sat}}}. \tag{5.23}$$

As r_v and r_c are updated by $\dot{r}^c_{\text{cond}}$, we also update θ_{vd} using the right hand side of (5.2b).

*In other formulations using p or Π as prognostic variables, the C-grid staggering behaviour is obtained by having p and Π in the same space as the density. The temperature T would be in the same space as θ_{vd} or any other temperature-like variable and so we do not need to denote it with a tilde here.

5.4.3 Coalescence of Cloud Water into Rain

This process consists of two parts: accretion of cloud water and the auto-accumulation of rain water. Following [87], we parametrise these processes with:

$$\dot{r}^0_{\text{accr}} = k_1(r_c - a) \tag{5.24}$$

and

$$\dot{r}^0_{\text{accu}} = k_2 r_c r_r^b \tag{5.25}$$

where $k_1 = 0.001\ \text{s}^{-1}$, $k_2 = 2.2\ \text{s}^{-1}$, $a = 0.001\ \text{kg kg}^{-1}$ and $b = 0.875$, which are the parameters used by [88]. As with the condensation/evaporation process for cloud water described in Section 5.4.2, this process is limited to prevent negative cloud water values forming.

5.4.4 Evaporation of Rain

The parametrisation we use for the evaporation of rain again comes from [87]. Having recovered ρ_d into the temperature space as $\widetilde{\rho}_d$, the evaporation rate that we use is

$$\dot{r}^r_{\text{evap}} = \frac{1}{\widetilde{\rho}_d}\frac{(1 - r_v/r_{\text{sat}}(\widetilde{p}, T))C(\widetilde{\rho}_d r_r)^{0.525}}{5.4 \times 10^5 + 2.55 \times 10^6/(\widetilde{p} r_{\text{sat}}(\widetilde{p}, T))}, \tag{5.26}$$

where $r_{\text{sat}}(\widetilde{p}, T)$ is taken from (5.23) and C is known as the ventilation factor:

$$C = 1.6 + 124.9(\widetilde{\rho}_d r_r)^{0.2046}. \tag{5.27}$$

As with the previous schemes, we limit the evaporation rate to prevent negative rain concentrations occurring. As with the condensation/evaporation scheme for cloud water, θ_{vd} is updated from (5.2b).

5.4.5 Sedimentation of Rain

Our approach to parametrising the fallout of rain is similar to the single moment scheme described in [89]. We assume that the number $n_r(D)$ of raindrops of diameter D forms a spectrum described by a Gamma distribution:

$$n_r(D) = \frac{N_r \lambda^{1+\mu}}{\Gamma(1+\mu)} D^\mu e^{-\lambda D}, \tag{5.28}$$

where N_r, μ and λ are parameters characterising the shape of the spectrum. N_r represents the total number density of particles, whilst μ and λ are known as the shape and inverse scale parameters. The Gamma function $\Gamma(1 + \mu)$ appears here to normalise the distribution. Assuming spherical raindrops, the mass of a raindrop of diameter D is

$$m_r(D) = \frac{\pi}{6}\rho_w D^3, \tag{5.29}$$

where $\rho_w = 1000$ kg m^{-3} is the density of liquid water. As the mass of a raindrop is proportional to the cube of D, the third moment of $n_r(D)$ represents the mass density of rain. We will also parametrise the terminal velocity of a falling raindrop of diameter D with another Gamma distribution, using a, b, f, ρ_0 and g as parameters.

$$V(D) = aD^b e^{-fD}\left(\frac{\rho_0}{\rho_d}\right)^g. \tag{5.30}$$

The term in the brackets represents a *density correction*.

Now we will link this distribution to our rainfall mixing ratio r_r. The mass density of rain in some area is given by $\rho_d r_r$, but we can find an equivalent expression from the diameter distribution. Equating these gives

$$\rho_d r_r = \int_0^\infty m_r(D) n_r(D)\, dD = \frac{\pi\rho_w N_r \Gamma(4+\mu)}{6\lambda^3\Gamma(1+\mu)}, \tag{5.31}$$

having performed the integral using properties of the Gamma function. This can be rearranged to obtain λ as a function of ρ_d and r_r:

$$\lambda = \left(\frac{\pi\rho_w N_r \Gamma(4+\mu)}{6\rho_d r_r \Gamma(1+\mu)}\right)^{\frac{1}{3}}. \tag{5.32}$$

The velocity of the rain mass density is given by

$$v_r = \frac{1}{\rho_d r_r}\int_0^\infty V(D) m_r(D) n_r(D)\, dD = \frac{\Gamma(4+\mu+b) a \lambda^{4+\mu}}{\Gamma(4+\mu)(\lambda+f)^{4+\mu+b}}\left(\frac{\rho_0}{\rho_d}\right)^g, \tag{5.33}$$

having performed the integral again using properties of the Gamma distribution. This value of v_r is then used to advect r_r vertically downwards in order to simulate rainfall.

The strategy that we use in our model is to first find λ from ρ_d and r_r. At this stage, no special care is taken with ρ_d. We then find v_r, projecting it into the velocity space V_v. The rain field is then advected by $-v_r\widehat{\mathbf{k}}$ using the same advection scheme used for r_r in the dynamics part of the model. In the $k = 0$ configuration, this is the recovered advection scheme presented in Chapter 4, whilst the $k = 1$ configuration uses the embedded DG scheme of [67]. To allow for fallout of the rain through the bottom of the domain, the upwind DG equation is modified from equation (3.12) to finding the r_{trial}, for all $\psi \in V_\theta$

that satifies

$$
\begin{aligned}
&\int_\Omega \psi r_{\text{trial}}\, dx - \int_\Omega \psi r_r\, dx + \Delta t \int_\Omega r_r \left[\boldsymbol{\nabla}\cdot\left(\psi v_r \widehat{\boldsymbol{k}}\right)\right]\, dx \\
&-\Delta t \int_{\partial\Omega} \left(v_r \widehat{\boldsymbol{k}}\cdot\widehat{\boldsymbol{n}}\right) r_r^\dagger \psi\, ds - \Delta t \int_\Gamma \left(v_r \widehat{\boldsymbol{k}}\cdot\widehat{\boldsymbol{n}}^+\right) r_r^\dagger [\![\psi]\!]_+\, dS = 0,
\end{aligned}
\tag{5.34}
$$

with $-v_r\widehat{\boldsymbol{k}}$ taking the role of $\overline{\boldsymbol{u}}$ from equation (3.12). This advection can also be limited.

For the parameters we use the following values: $N_r = 10^5$, $\mu = 0$, $a = 362$, $b = 0.65$, $f = 0$, $g = 0.5$, $\rho_0 = 1.22$ kg m^{-3}. There is a singularity in the expression for λ at $r_r = 0$ (which could be the case for much of the domain). We therefore only call the scheme for $r_r > 10^{-10}$.

5.4.6 Time Discretisation

The time integration currently used for the parametrisations is a simple first-order explicit scheme. In other words, for a process $\dot{r}$ affecting a variable r, the new value r_{new} will be related to the old r_{old} by

$$
r_{\text{new}} = r_{\text{old}} + \Delta t\, \dot{r}(r_{\text{old}}). \tag{5.35}
$$

The exception here is the treatment of the sedimentation of rainfall, which was described in the previous section. The value of $\dot{r}$ comes from the state of the model just before this physics routine is called. This is the state of the model after either the dynamics or the preceding physics routine has been completed.

5.5 Setting Up Hydrostatic Balance

For many test cases the background or initial state of the model will be in hydrostatic balance. In Section 3.2.8 we presented the strategy for finding the ρ_d corresponding to a θ_d profile that yields a discrete hydrostatic balance. As the model is extended to include moisture, we extend this procedure to facilitate model initial conditions that are not given in terms of our prognostic variables, as for the test cases in Section 5.6.

We will consider cases in which thermodynamic and moist variables are provided and the prognostic variables including the density ρ_d need to be found. These will involve splitting the process into two steps which are iterated until the state has converged to within some specified tolerance. The first step is to use the best guesses of the thermodynamic prognostic variables θ_{vd}, r_v and r_c to calculate the density ρ_d that provides hydrostatic balance given those values of θ_{vd}, r_v and r_c. This uses the routine described in Section 3.2.8, finding the $(\boldsymbol{w}, \rho_d) \in \left(V_v^{\text{vert}}, V_\rho\right)$ that solve, for all $(\boldsymbol{\psi}, \lambda) \in \left(V_v^{\text{vert}}, V_\rho\right)$, the coupled equations

$$\begin{aligned}\int_{\Omega}\boldsymbol{\psi}\cdot\boldsymbol{w}\,\mathrm{d}x-\int_{\Omega}c_{pd}\Pi(\rho_d,\theta_{vd})\boldsymbol{\nabla}\cdot\left(\frac{\theta_d\boldsymbol{\psi}}{1+r_t}\right)=&\\ -\int_{\partial\Omega_i}\frac{c_{pd}\theta_{vd}}{1+r_t}\Pi_R\boldsymbol{\psi}\cdot\widehat{\boldsymbol{n}}\,\mathrm{d}s-\int_{\Omega}g\boldsymbol{\psi}\cdot\widehat{\boldsymbol{k}}\,\mathrm{d}x,&\end{aligned} \tag{5.36a}$$

$$\int_{\Omega}\lambda\boldsymbol{\nabla}\cdot\boldsymbol{w}\,\mathrm{d}x=0, \tag{5.36b}$$

provided a prescribed Π_R on either the top or bottom boundary $\partial\Omega_i$. This differs from equation (3.26) only by the inclusion of the moisture species.

The second step is to use the best guess of ρ_d and the specified initial conditions to find θ_{vd}, r_v and r_c, where we shall assume that there is no initial rain. The second step is considerably simpler when the variables specified in the initial conditions are one of our prognostic variables. We will consider two cases of initial conditions, with saturated and unsaturated atmospheres.

5.5.1 Saturated Conditions

This set-up involves initial conditions like those in the moist benchmark of [78]. The problem is to find ρ_d, θ_{vd} and r_v given θ_e, r_t and a boundary condition on the pressure. Assuming the absence of rain, $r_v = r_{\text{sat}}$, and $r_c = r_t - r_v$. The *wet-equivalent potential temperature*, θ_e, is a conserved quantity in reversible, moist adiabatic processes, i.e. $D\theta_e/Dt = 0$. Following [90], for our equation set (5.2), θ_e can be written as

$$\theta_e := T\left(\frac{p_0}{p}\right)^{\frac{R_d}{c_{pd}+c_{pl}r_t}}(\mathcal{H})^{\frac{-r_vR_v}{c_{pd}+c_{pl}r_t}}\exp\left[\frac{L_v(T)r_v}{(c_{pd}+c_{pl}r_t)T}\right], \tag{5.37}$$

which with $\mathcal{H}=1$ for a saturated atmosphere is the same as that used by [78] and that derived in the appendix of [91] from the second law of thermodynamics.

The challenge is to obtain the θ_{vd} and ρ_d fields that satisfy the specified θ_e field whilst ensuring that $r_v = r_{\text{sat}}$. We use an initial guess for θ_{vd} and feed it to equation (5.36), to generate a guess for ρ_d. This density is then converted into V_θ, before a nested fixed-point iteration-style procedure is used to invert $\theta_e(\theta_{vd},\widetilde{\rho}_d,r_v)$ and $r_{\text{sat}}(\theta_{vd},\widetilde{\rho}_d,r_v)$, to obtain θ_{vd} and r_v.

Let l, m and n count the number of iterations to find ρ_d, θ_{vd} and r_v respectively. These form nested loops, such that the outer loop uses the latest approximations of θ_{vd} and r_v with equation (5.36) to obtain $\rho_d^{(l+1/2)}$. We then update the approximation to ρ_d, combining this with the previous value to provide stability, doing

$$\rho_d^{(l+1)} = (1-\delta)\rho_d^{(l)} + \delta\rho_d^{(l+1/2)}, \tag{5.38}$$

where we chose $\delta = 0.8$. The next loop finds θ_{vd} using

$$\theta_{vd}^{(m+1/2)} = \frac{\theta_{vd}^{(m)}\theta_e}{\theta_e\left(\theta_{vd}^{(m)}, \widetilde{\rho}_d^{(l)}, r_v^{(n)}\right)}, \quad \theta_{vd}^{(m+1)} = (1-\delta)\theta_{vd}^{(m)} + \delta\theta_{vd}^{(m+1/2)}, \tag{5.39}$$

whilst the inner loop computes

$$r_v^{(n+1/2)} = r_{\text{sat}}\left(\theta_{vd}^{(m)}, \widetilde{\rho}_d^{(l)}, r_v^{(n)}\right), \quad r_v^{(n+1)} = (1-\delta)r_v^{(m)} + \delta r_v^{(m+1/2)}, \tag{5.40}$$

where θ_e without arguments denotes the specified value. These loops are iterated until $\theta_e(\theta_{vd}, \widetilde{\rho}_d, r_v)$ converges to the specified value to some tolerance.

5.5.2 Unsaturated Conditions

We now discuss how to find the prognostic thermodynamic variables given θ_d and the relative humidity $\mathcal{H}$, such as the case in Section 5.6.3 or [92]. The relative humidity is related to r_v by

$$\mathcal{H} = \frac{r_v}{r_{\text{sat}}}\left(\frac{1 + r_{\text{sat}}/\epsilon}{1 + r_v/\epsilon}\right). \tag{5.41}$$

As in Section 5.5.1, we use what can be thought of as a nested iterative procedure. Counting the latest approximations of ρ_d, θ_{vd} and r_v with l, m and n, the outer loop uses $\theta_{vd}^{(m)}$ and $r_v^{(n)}$ in the hydrostatic balance equation (5.36) to determine $\rho_d^{(l+1/2)}$. Again, the next value of ρ_d is given by

$$\rho_d^{(l+1)} = (1-\delta)\rho_d^{(l)} + \delta\rho_d^{(l+1/2)}. \tag{5.42}$$

There is only one inner loop in this case, so that $m = n$. The new value of r_v is found by rearranging (5.41) so that

$$r_v^{(m+1/2)} = \frac{\mathcal{H}r_{\text{sat}}\left(\theta_{vd}^{(m)}, \widetilde{\rho}_d^{(l)}, r_v^{(m)}\right)}{1 + (1-\mathcal{H})r_{\text{sat}}\left(\theta_{vd}^{(m)}, \widetilde{\rho}_d^{(l)}, r_v^{(m)}\right)/\epsilon}, \tag{5.43a}$$

$$r_v^{(m+1)} = (1-\delta)r_v^{(m)} + \delta r_v^{(m+1/2)}, \tag{5.43b}$$

where $\mathcal{H}$ is the specified value of the relative humidity. The final step is to use the specified value of θ_d to get

$$\theta_{vd}^{(m)} = \theta_d\left(1 + r_v^{(m)}/\epsilon\right). \tag{5.44}$$

This iterative process continues until $\mathcal{H}\left(\theta_{vd}^{(m)}, \rho_{d}^{(l)}, r_{v}^{(m)}\right)$ has converged to its specified value to some tolerance.

5.6 Test Cases

In this section we demonstrate the discretisation detailed in previous sections through a series of test cases, with some comparison of the $k = 0$ and $k = 1$ configurations of the model. Two new test cases are presented, featuring a gravity wave in a saturated atmosphere and a three-dimensional rising thermal in a saturated atmosphere. These new tests are introduced in part to showcase the capability of our model, but also to help address the lack of commonly used test cases between simple dry test cases and full-physics simulations of the atmosphere. No artificial diffusion is included in any of these tests.

5.6.1 Bryan and Fritsch Moist Benchmark

The first demonstration of our discretisation is through the moist benchmark test case of [78], which simulates a rising thermal through a cloudy atmosphere. This test captures the effects of latent heat in a relatively simple high-resolution two-dimensional model. Since the atmosphere is entirely cloudy, the motion is smooth, allowing different models to be easily compared. The domain is a vertical slice of width $L = 20$ km and height $H = 10$ km. Periodic boundary conditions are applied at the vertical boundaries, but the top and bottom boundaries are rigid, so that $\boldsymbol{v} \cdot \widehat{\boldsymbol{n}} = 0$ along them. As in [78], we include no rain microphysics and no Coriolis force.

The initial conditions defined in [78] specify a background state with constant $r_t = 0.02$ kg kg^{-1} and constant wet-equivalent potential temperature $\theta_e = 320$ K, which is defined in (5.37). Along with these, the background state is given by the requirements of hydrostatic balance, $r_v = r_{\text{sat}}$ and $p = 10^5$ Pa at the bottom boundary. The procedure described in Section 5.5.1 allows us to obtain the prognostic variables θ_{vd}, ρ_d, r_v and r_c that approximately satisfy these conditions.

A perturbation is then applied to θ_{vd}. With (x, z) as the horizontal and vertical coordinates, the perturbed field is

$$\theta'_{vd} = \begin{cases} \Delta\Theta \cos^2\left(\frac{\pi r}{2 r_c}\right), & r < r_c, \\ 0, & \text{otherwise,} \end{cases} \tag{5.45}$$

where $\Delta\Theta = 2$ K, $r_c = 2$ km and with $x_c = L/2$ and $z_c = 2$ km we define

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}. \tag{5.46}$$

The initial θ_{vd} field is given in terms of the background field $\overline{\theta}_{vd}$:

$$\theta_{vd} = \overline{\theta}_{vd}\left(1 + \frac{\theta'_{vd}}{300\text{ K}}\right). \tag{5.47}$$

In the test case of [78], the pressure field is unchanged by the perturbation. As in Section 4.6.1, we replicate this by finding ρ_d such that for all $\zeta \in V_\rho$,

$$\int_\Omega \zeta\rho_d\theta_{vd}\,dx = \int_\Omega \zeta\overline{\rho}_d\overline{\theta}_{vd}\,dx, \tag{5.48}$$

where $\overline{\rho}_d$ and $\overline{\theta}_{vd}$ are the background states for ρ_d and θ_{vd}. The system is returned to saturation by finding the r_v that solves, for all $\phi \in V_\theta$,

$$\int_\Omega \phi r_v\,dx = \int_\Omega \phi r_{sat}(\widetilde{\rho}_d, \theta_{vd}, r_v)\,dx, \tag{5.49}$$

where $r_{sat}(\widetilde{\rho}_d, \theta_{vd}, r_v)$ is the expression for saturation mixing ratio in terms of the initial ρ_d recovered into V_θ and the initial θ_{vd} field that have already been found, and also the unknown r_v to be solved for. Finally, r_c is found from applying $r_c = r_t - r_v$. The initial velocity field is zero in each component.

Figures 5.3 and 5.4 show the θ_e and vertical velocity w fields at $t = 1000$ s. Figure 5.3 shows these fields for the configuration using the $k = 0$ lowest-order spaces, with the $k = 1$ spaces shown in Figure 5.4. Both simulations used $\Delta x = \Delta z = 100$ m and $\Delta t = 1$ s. These final states are visibly different: whilst the $k = 0$ solutions resemble those of [78], the $k = 1$ solution displays an extra plume forming at the top of the rising thermal. We believe this to be a manifestation of a physical instability that is damped by numerical diffusion in the $k = 0$ case. The $k = 1$ solution appears highly sensitive to the choice of mesh, as at higher resolution the θ_e field does not appear to converge to a single solution. Indeed if the domain is spanned horizontally by an odd number of cells, rather than a secondary plume emerging, the top of the primary plume appears to collapse. This behaviour is also observed in the absence of moisture.

5.6.2 Inertia-Gravity Waves in Saturated Atmosphere

We present here a new test case, a moist version of the non-hydrostatic gravity wave test of [82], but in a saturated atmosphere like that of the moist benchmark in [78]. Without strong vertical motions, this test should avoid the issues with the rising plume of the test from Section 5.6.1. The final state of this test is spatially smooth, making this test appropriate for convergence tests. No rain physics is used in this test case, and there is also no Coriolis force.

The problem is set up in a two-dimensional vertical slice of length $L = 300$ km and

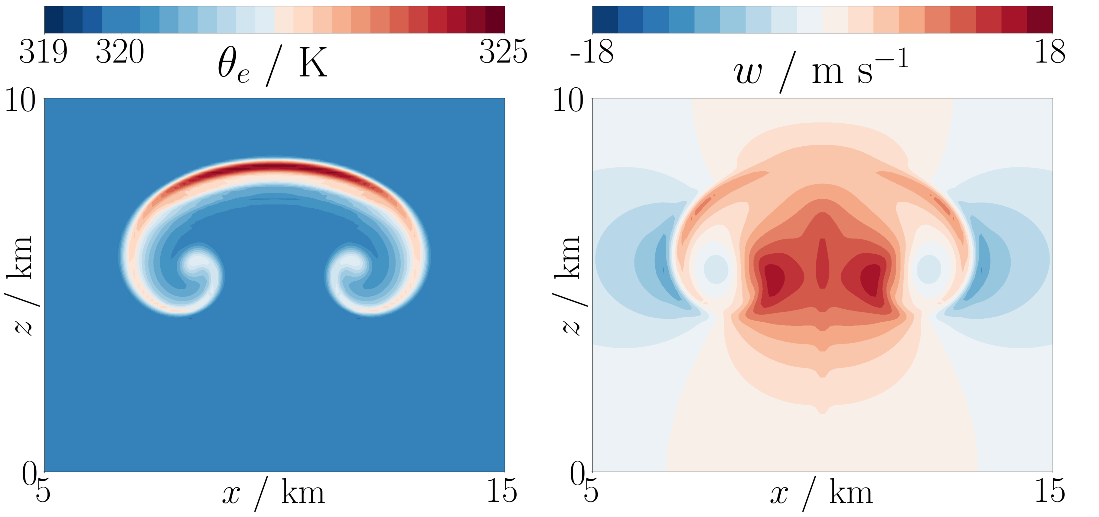


Figure 5.3: The (left) θ_e field contoured every 0.25 K and (right) vertical velocity w field contoured every 2 m s^{-1}, with both fields plotted at $t = 1000$ s for a simulation of the moist benchmark case [78] representing a thermal rising through a saturated atmosphere. The 320 K contour has been omitted for clarity in the θ_e field. This simulation is with the lowest-order $k = 0$ set of spaces, with grid spacing $\Delta x = \Delta z = 100$ m and a time step of $\Delta t = 1$ s. This test is a common milestone in the development of moist atmospheric models because its solution is comparable to the dry case and because the cloudy atmosphere is neutrally stable, unlike many other moist rising thermal tests. It also demonstrates the performance at relatively high resolution compared with numerical weather prediction models. However setting up the initial conditions can be challenging, as they are prescribed in terms of θ_e which is not usually a prognostic variable. These solutions are visibly similar to those presented in [78].

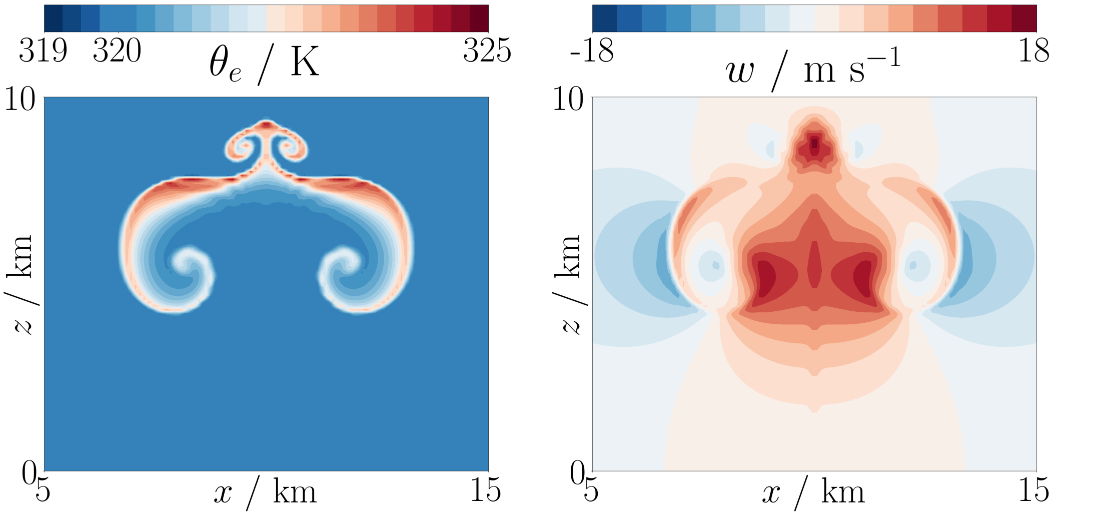


Figure 5.4: Outputted fields from the $k = 1$ next-to-lowest order space simulation at $t = 1000$ s of the moist benchmark from [78]. (Left) θ_e with contours spaced by 0.25 K and (right) vertical velocity w contoured every 2 m s^{-1}. The simulation used grid spacing $\Delta x = \Delta z = 100$ m and a time step of $\Delta t = 1$ s. The 320 K contour has been omitted for clarity in the θ_e field. A second plume can be seen forming at the top of the primary plume, in contrast to the results in Figure 5.3 with the $k = 0$ spaces. Although the secondary plume is not typically reported in the literature, we believe that rising thermal test cases like this do admit multiple solutions, and that this secondary plume is then triggered by dynamic instabilities as the leading edge of the bubble involves a very steep temperature gradient. However in many models, including the one we present with the lowest-order spaces, the advection schemes can damp this instability causing the smooth results seen in Figure 5.3.

height $H = 10$ km. The dry gravity wave setup used in [82] applies a perturbation to θ_d, which has a stratified background profile and is in hydrostatic balance. Our variation on this is to apply the perturbation to the stratified background profile of θ_e in hydrostatic balance. Using (x, z) as the horizontal and vertical coordinates, the specified θ_e profile is

$$\overline{\theta}_e = \Theta_0 e^{N^2 z/g}, \tag{5.50}$$

where $\Theta_0 = 300$ K and $N^2 = 10^{-4}$ s^{-2}. With $r_t = 0.02$ kg kg^{-1} everywhere and the boundary condition of $p = 10^5$ Pa at $z = 0$, we use the hydrostatic balance procedure laid out in Section 5.5.1 to find the θ_{vd}, ρ_d, r_v and r_c fields that correspond to these initial conditions with the requirements of hydrostatic balance and that $r_v = r_{sat}$ everywhere. The initial velocity applied is $\boldsymbol{v} = (U, 0)$ with $U = 20$ m s^{-1} describing uniform flow in the x-direction. This defines each of the mean fields.

A perturbation is then added, which is specified as

$$\theta_e' = \frac{\Delta\Theta}{1 + a^{-2}(x - L/2)^2} \sin\left(\frac{\pi z}{H}\right), \tag{5.51}$$

with $a = 5 \times 10^3$ m and $\Delta\Theta = 0.01$ K. The perturbed initial condition is then given by $\theta_e = \overline{\theta}_e + \theta_e'$. Setting the new requirements that both r_t and the pressure are unchanged by the addition of the perturbation, and that we still have $r_v = r_{sat}$, defines the problem necessary to solve to find the initial prognostic fields. We do this via a nested iterative process related to that described in Section 5.5.1. In the outer loop we find ρ_d^h such that for all $\zeta \in V_\rho$

$$\int_\Omega \zeta \rho_d^h \theta_{vd}^n \, dx = \int_\Omega \zeta \overline{\rho}_d \overline{\theta}_{vd} \, dx, \tag{5.52}$$

which is combined with the previous best estimate of ρ_d^n to give $\rho_d^{n+1} = (1 - \delta)\rho_d^n + \delta \rho_d^h$, where $\delta = 0.8$. Nested inside this process are more damped iterations to find θ_{vd} and r_v, exactly as in Section 5.5.1.

Figure 5.5 shows the perturbation to the final diagnostic θ_e field at $t = 3600$ s for simulations with the $k = 0$ lowest-order spaces with $\Delta x = \Delta z = 500$ m and the $k = 1$ set of spaces, where $\Delta x = \Delta z = 1000$ m, both with $\Delta t = 1.2$ s. These different cases are not visibly different from one another, and closely resemble the final state of the dry case from [82].

Convergence

As the solution and evolution of this example is spatially smooth, we can use it to form a convergence test upon our model. Although the final state (at $t = 3600$ s) does not have an analytic solution, we use the fields from a high resolution simulation as the true

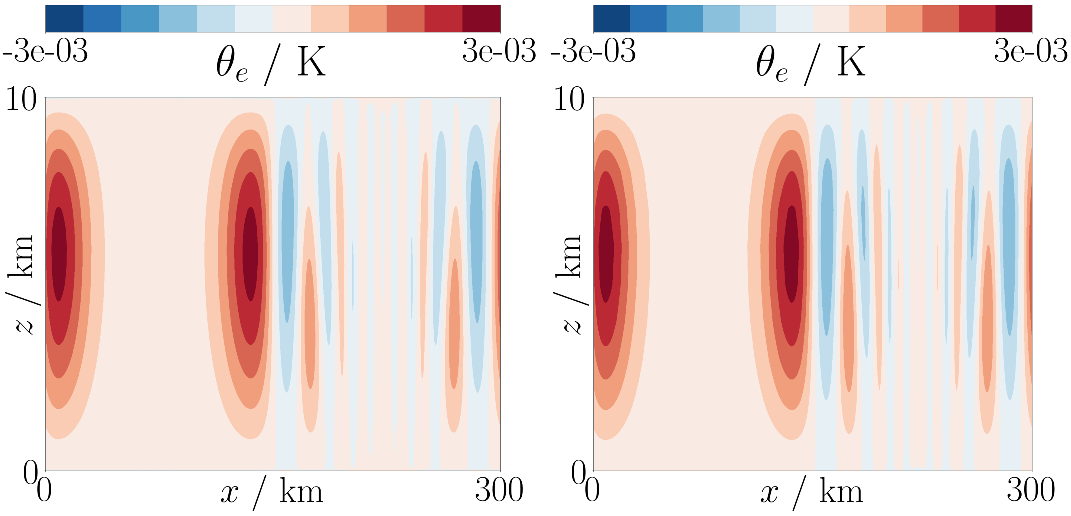


Figure 5.5: The perturbations to the θ_e fields at $t = 3600$ s for the moist gravity wave test case. (Left) the $k = 0$ lowest-order spaces set-up using $\Delta x = 500$ m and (right) the $k = 1$ spaces with $\Delta x = 1000$ m. Both cases used $\Delta t = 1.2$ s. Contours are spaced every 5×10^{-4} K. This test case represents a cloudy-atmosphere variant of the dry gravity wave test of [82] and is more gentle than the rising thermal test of Section 5.6.1. Its solution is spatially smooth, allowing this test to be used to investigate the spatial convergence properties of the model.

solution.

To measure the spatial accuracy of our model, we ran this test case at different resolutions, with each using a time step of $\Delta t = 1.2$ s, which is small enough to avoid the Courant number breaching its critical value in the highest resolution cases. The error is measured looking at the θ_e diagnostic in V_θ at $t = 3600$ s. The θ_e fields are interpolated onto the finest mesh, which has $\Delta x = 100$ m for the $k = 0$ case but $\Delta x = 200$ m for the $k = 1$ case. The error between the high resolution solution for θ_e and those run at coarser resolutions is measured in the L^2 norm. Results for our model are plotted in Figure 5.6, which indicates that in both the $k = 0$ and $k = 1$ cases the model has second-order spatial accuracy, with the error proportional to $(\Delta x)^2$.

5.6.3 Rising Thermal with Rain

This test case is based upon one described in [92]. This involves a thermal rising in an unsaturated atmosphere, forming a cloud which rains out, thus demonstrating the description of rain within our model. This is another two dimensional vertical slice test, this time with domain of height $H = 2.4$ km and length $L = 3.6$ km, again with periodic conditions at the vertical sides. The Coriolis force is neglected.

In contrast to the saturated atmosphere initial conditions of Sections 5.6.1, the initial state is defined by the dry potential temperature θ_d and a relative humidity field $\mathcal{H}$. The

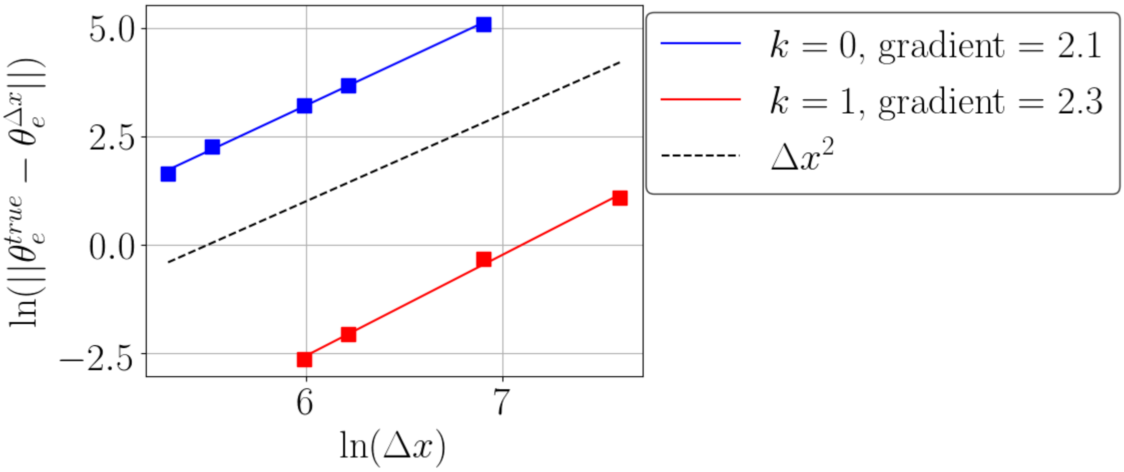


Figure 5.6: A convergence plot showing the error as a function of resolution in the final state from the moist gravity wave test of Section 5.6.2. The true solution was taken from a high resolution simulation. Both the $k = 0$ and $k = 1$ configurations of the model appear to have second-order or better accuracy, which is one of the properties targeted by [9].

background fields are $\mathcal{H} = 0.2$ and

$$\theta_d = \Theta e^{Sz}, \tag{5.53}$$

where Θ is the dry potential temperature corresponding to $T_{surf} = 283$ K and $p = 8.5\times10^4$ Pa, which also provides the pressure condition at the boundary. The stratification is given by $S = 1.3 \times 10^{-5}$ m^{-1}. We then use the procedure outlined in Section 5.5.2 to find the background θ_{vd}, ρ_d and r_v fields that satisfy hydrostatic balance. The initial r_c and r_r fields are zero.

The perturbation is then applied to the relative humidity field $\mathcal{H}$, with a circular bubble that is just saturated, with an outer disk smoothing the perturbation into the background state. This initial relative humidity field is given by

$$\mathcal{H} = \begin{cases} \overline{\mathcal{H}}, & r \geqslant r_1, \\ \overline{\mathcal{H}} + (1 - \overline{\mathcal{H}})\cos^2\left(\frac{\pi(r-r_2)}{2(r_1-r_2)}\right), & r_2 \leqslant r < r_1, \\ 1, & r < r_2, \end{cases} \tag{5.54}$$

where $\overline{\mathcal{H}} = 0.2$ and

$$r = \sqrt{(x - x_c)^2 + (z - z_c)^2}, \tag{5.55}$$

with $x_c = L/2$, $z_c = 800$ m, $r_1 = 300$ m and $r_2 = 200$ m. The r_v and θ_{vd} that correspond to this $\mathcal{H}$ are found via a fixed point iterative method.

Fields are displayed in Figures 5.7 and 5.8 for θ'_{vd} and r_r at $t = 300$ and 600 s. Both simulations use the $k = 0$ lowest-order space set-up, with $\Delta x = 20$ m and $\Delta t = 1$ s. Figure 5.7

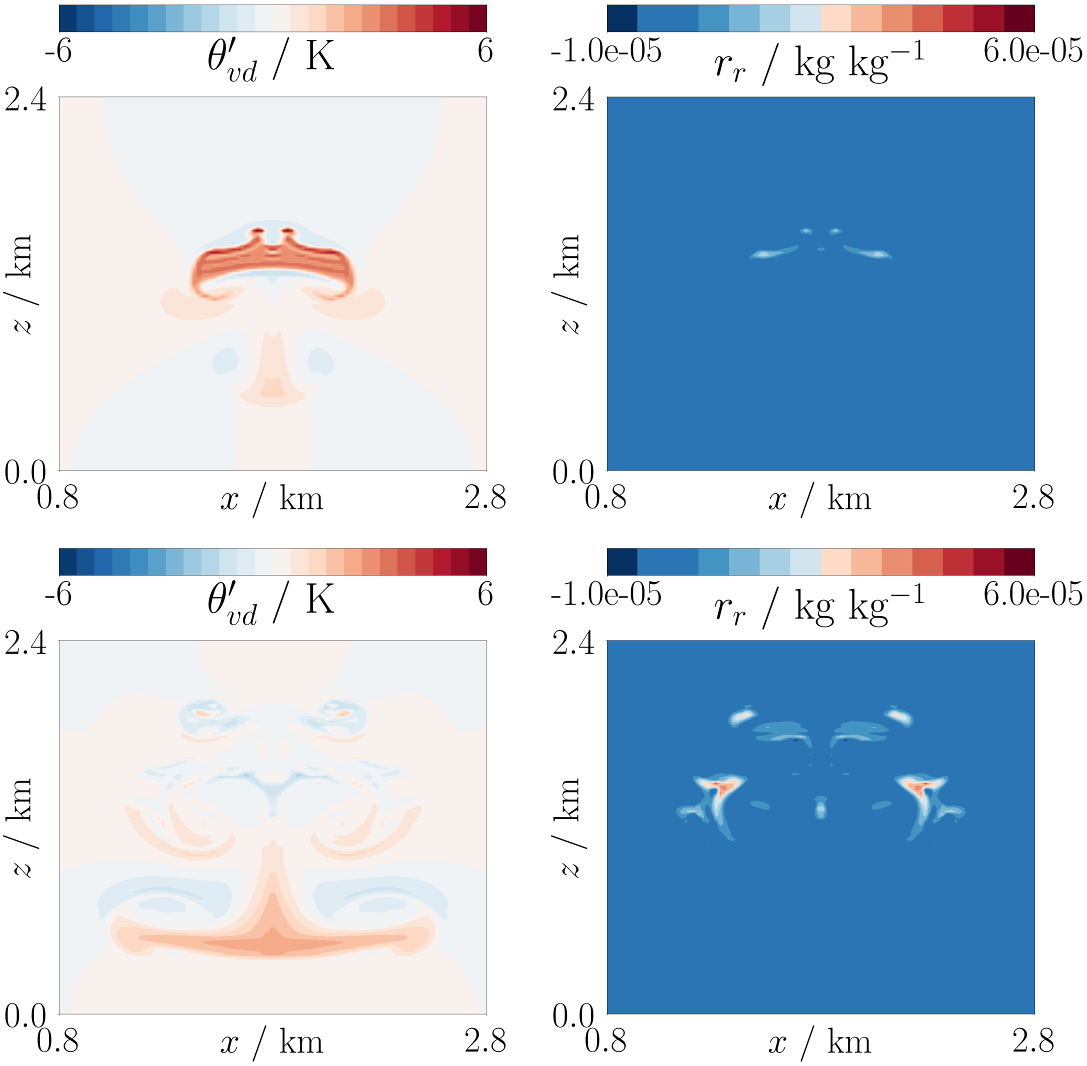


Figure 5.7: The field output at (top row) $t = 300$ s and (bottom row) $t = 600$ s from the rising thermal test with rain. Fields shown are in the $k = 0$ spaces with $\Delta x = \Delta z = 20$ m and a time step of $\Delta t = 1$ s. (Left) the perturbation to θ_{vd}, contoured every 0.25 K. (Right) the rain field r_r, with contours every 5×10^{-6} kg kg^{-1} and omitting the zero contour. This test case is initially unsaturated, but as the thermal rises, water vapour condenses to form a cloud which then rains out, demonstrating the use of rain within our model.

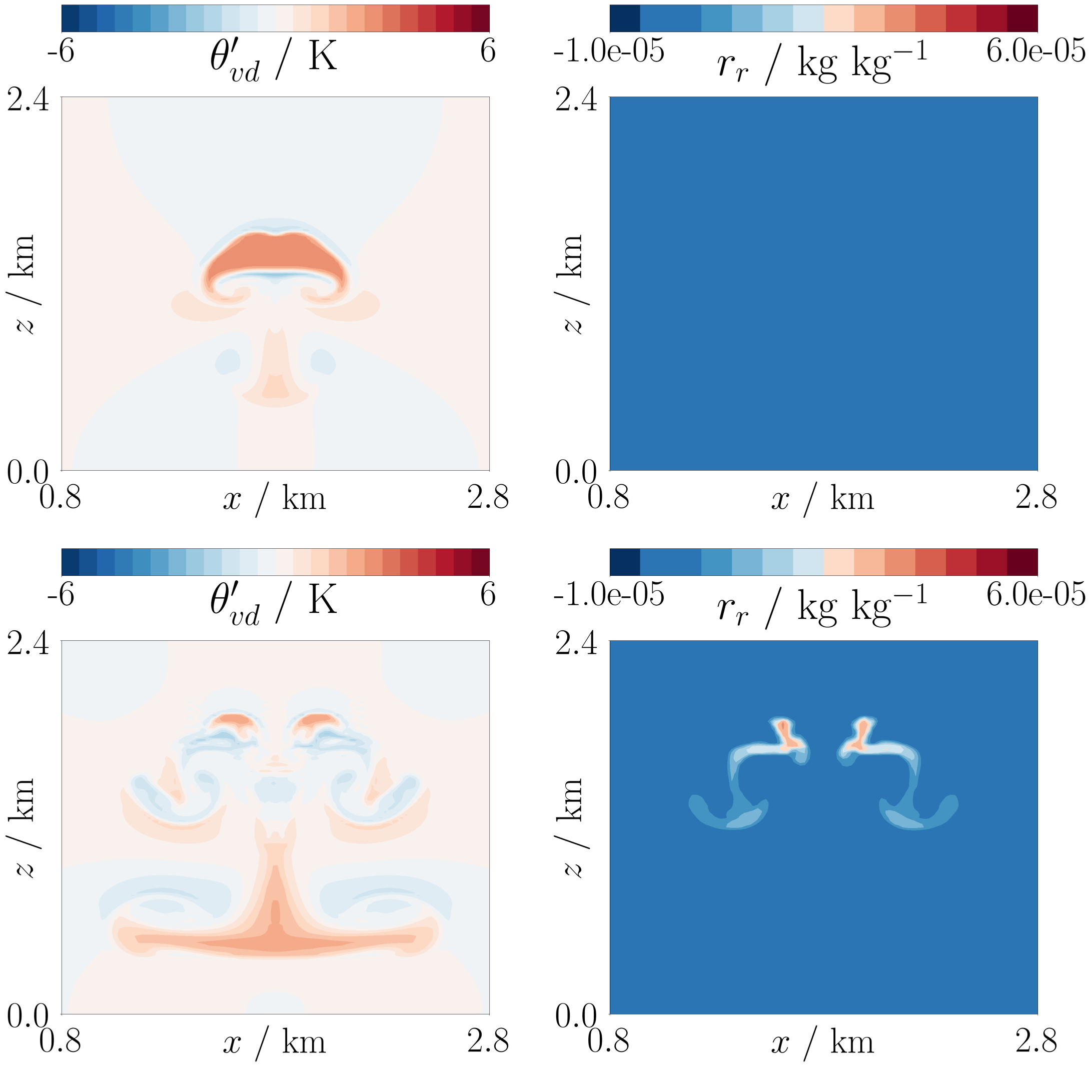


Figure 5.8: The same results as in Figure 5.7, but with a limiter applied to the moisture species. The θ_{vd} solution is notably smoother and lower cloud concentrations form, leading to less rain which also begins later.

shows the results with no limiter applied to the advected moisture fields, whilst Figure 5.8 shows the same set-up but with a limiter applied to all the moisture variables.

The length scales of the simulation are small enough for it to be highly turbulent, with the final state dependent on the resolution in the absence of a turbulence parametrisation. Indeed, the lack of turbulence parametrisation in our model explains why these results look significantly different to those of [92]. Comparing Figures 5.7 and 5.8 demonstrates the effect of limiting the transport of moisture species. In the absence of the limiter, the mass of water vapour depreciates less and so more cloud is formed, associated with a greater release of latent heat and a stronger updraught. We see that rain forms earlier in the absence of a limiter. However, some negative moisture values do form, which are absent from the limited case.

5.6.4 Three-Dimensional Thermal in a Saturated Atmosphere

We now demonstrate the use of our discretisation upon small-scale dynamics in three dimensions. This test case is a three-dimensional version of the moist benchmark of [78] that was described in Section 5.6.1. Rain and the effects of planetary rotation are not included.

The domain is now periodic in the horizontal directions, with length, width and height 10 km. The background state set-up is the same as that in Section 5.6.1, with $\theta_e = 320$ K, $r_t = 0.02$ kg kg^{-1} and the pressure as $p = 10^5$ Pa on the bottom surface. Using the initialisation procedure that was outlined in Section 5.5.1 generates the values of the prognostic variables such that the model is in hydrostatic balance and saturated with respect to water vapour.

Using $x_c = y_c = 5$ km and $z_c = 2$ km, and defining

$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2}, \tag{5.56}$$

we apply the perturbation

$$\theta'_{vd} = \begin{cases} \Delta\Theta \cos^2\left(\frac{\pi r}{2r_c}\right), & r < r_c, \\ 0, & \text{otherwise}, \end{cases} \tag{5.57}$$

with $\Delta\Theta = 1$ K and $r_c = 2$ km. As in Section 5.6.1, the perturbation is applied using the background field $\bar{\theta}_{vd}$

$$\theta_{vd} = \bar{\theta}_{vd}\left(1 + \frac{\theta'_{vd}}{300\text{ K}}\right). \tag{5.58}$$

The same routine as used in Section 5.6.1 is then applied to obtain the initial ρ_d, r_v and r_c fields, ensuring that the atmosphere is exactly saturated and that the initial pressure

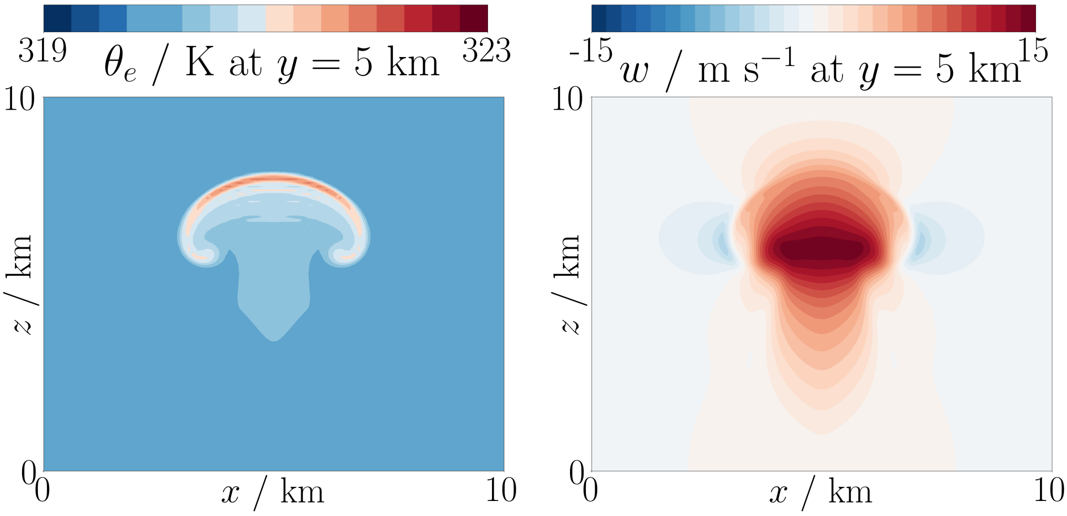


Figure 5.9: The (left) θ_e and (right) vertical velocity w fields at $t = 1000$ s for the three-dimensional simulation of a thermal rising through a saturated atmosphere. The θ_e field is contoured every 0.25 K with the 320 K contour omitted, whilst the contour spacing for the w field is 1 m s^{-1}. Cross-sections are shown at $y = 5$ km, with values plotted on the lower side of the plane. This simulation is with the $k = 0$ lowest-order set of spaces, with grid spacing $\Delta x = 100$ m and a time step of $\Delta t = 1$ s. This test is introduced to demonstrate the model at high spatial resolutions in three dimensions. As a three-dimensional version of the test case in Section 5.6.1, it can also exploit the same initialisation code described in Section 5.5.1.

field is equal to the background pressure field.

Cross-sections of the θ_e and vertical velocity w fields at $t = 1000$ s and $y = 5$ km are shown in Figures 5.9 and 5.10, for both the set-ups using $k = 0$ lowest-order spaces (which had $\Delta x = \Delta y = \Delta z = 100$ m) and the $k = 1$ spaces (which had $\Delta x = \Delta y = \Delta z = 200$ m). Both simulations had $\Delta t = 1$ s. As in Section 5.6.1, we see a secondary plume beginning to form at the top of the rising thermal in the $k = 1$ case.

5.6.5 Moist Baroclinic Wave

The final test case that we present is the moist baroclinic wave outlined in the appendix of [79]. This is the only test featuring the Coriolis force, although rain is again neglected. It is a large-scale test, resembling conditions used in numerical weather prediction more than the other tests described in this Section, but as such it is more challenging. The set-up is very similar to that described in Section 4.6.2, in the same domain and using the same expressions for zonal wind u and geopotential Φ in terms of the (x, y, η) coordinates. The key differences are that the expression for temperature is replaced by an expression for virtual temperature T_v and the addition of an expression for specific

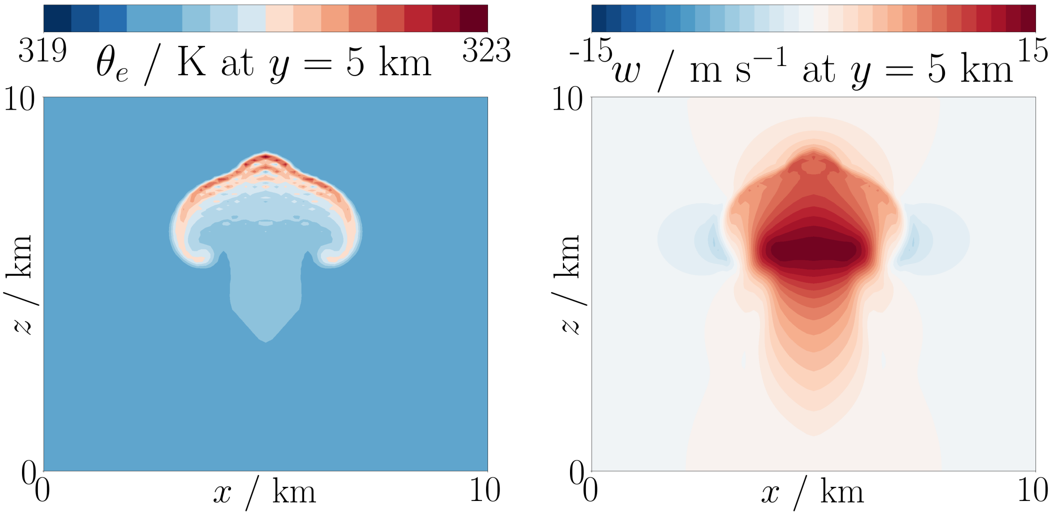


Figure 5.10: The outputted fields along $y = 5$ km from the $k = 1$ next-to-lowest order space simulation at $t = 1000$ s of the three-dimensional simulation of a rising thermal in a saturated atmosphere. (Left) θ_e and (right) vertical velocity w fields for a simulation using grid spacing $\Delta x = 200$ m and a time step of $\Delta t = 1$ s. The θ_e field is contoured every 0.25 K with the 320 K contour omitted, whilst the contour spacing for the w field is 1 m s^{-1}. Cross-sections are shown at $y = 5$ km, with values plotted on the lower side of the plane. As with the two-dimensional case, a second plume can be seen forming at the top of the primary plume. As was described in Figure 5.4, this is still believed to be a valid solution and may be appearing with the $k = 1$ elements if these advection schemes are less damping than those for the lowest-order elements.

humidity[†] q. These expressions are:

$$\begin{aligned} T_v = {} & T_0 \eta^{\frac{R_d \Gamma}{g}} \\ & + \frac{f_0 u_0}{2R_d}\left[y - \frac{W}{2}\left(1 + \frac{1}{\pi}\sin\left(\frac{2\pi y}{W}\right)\right)\right]\left[\frac{2}{b^2}(\ln\eta)^2 - 1\right]\exp\left[-\left(\frac{\ln\eta}{b}\right)^2\right], \end{aligned} \tag{5.59a}$$

$$q = \frac{q_0}{2}\exp\left[-\left(\frac{y}{\Delta y_w}\right)^4\right]\begin{cases} 1 + \cos\left[\frac{\pi(1-\eta)}{1-\eta_w}\right], & \eta \geqslant \eta_w, \\ 0, & \text{otherwise.} \end{cases} \tag{5.59b}$$

The new constants are $\Delta y_w = 3.2 \times 10^6$ m and $\eta_w = 0.3$, while all other constants take the same values as in Section 4.6.2. We use the slightly lower value of $q_0 = 0.016$ than [79] to prevent our model being initially too close to saturation. The virtual temperature T_v now replaces T in the Newton iteration procedure to find η, which becomes

$$\eta^{(n+1)} = \eta^{(n)} - \frac{\Phi\left(\eta^{(n)}\right) - gz}{T_v\left(\eta^{(n)}\right) - R_d/\eta^{(n)}}. \tag{5.60}$$

In order to convert T_v and q into θ_{vd} and r_v, we use the definitions of $\theta_{vd}, \eta = p/p_R$ and the relations

$$T = \frac{T_v}{1 + q(R_v/R_d - 1)} \tag{5.61}$$

[†] Note that neither of these thermodynamic variables are our prognostic variables.

and

$$r_v = \frac{q}{1-q}. \tag{5.62}$$

As before, these θ_{vd} and r_v fields are used to compute the ρ_d field that provides hydrostatic balance for the background state using the procedure outlined in Section 3.2.8, with $\Pi = 1$ on the bottom boundary. The same perturbation is added to the zonal velocity.

Figures 5.11 and 5.12 show fields from this test case at $t = 12$ days for the $k = 0$ and $k = 1$ spaces respectively. For the $k = 0$ configuration, $\Delta x = \Delta y = 200$ km and $\Delta z = 1$ km, whilst for $k = 1$ these were $\Delta x = \Delta y = 250$ km and $\Delta z = 1.5$ km. For both simulations, $\Delta t = 300$ s. In the $k = 1$ simulation, the baroclinic wave becomes much stronger than in the $k = 0$ simulation. As the wave develops, the maxima and minima in the temperatures in the $k = 1$ case are higher than for the $k = 0$ lowest-order spaces. When these minima coincide with regions close to water vapour saturation, more condensation occurs. This releases latent heat and strengthens the baroclinic wave, reinforcing the behaviour.

5.7 Summary and Outlook

In this chapter, we have presented a discretisation of the moist compressible Euler equations that uses a compatible finite element framework. As such, this chapter documents the first implementation of physics parametrisations within the compatible finite element framework of Gusto. Building upon the model described in Chapters 3 and 4, we first identified the dynamical parts of the model which required change for the inclusion of moisture, which involved amending the limiter of [75] for the V_θ space which is quadratic in the vertical direction. The coupling of the dynamical part of the model to the parametrisations was then discussed, addressing the problem of how to combine fields from different function spaces for use in parametrisations. The chosen method was to use the recovery operator outlined in Chapter 4 to obtain ρ_d within V_θ, as this possessed second-order accuracy due to its accurate representation of the field at the boundaries of the domain. After describing the microphysics parametrisations used in the model, strategies were presented for obtaining hydrostatic balance when provided initial conditions in terms of diagnostic (rather than prognostic) variables, which is important for setting up many test cases. The use of the discretisation was demonstrated through a series of test cases, and two new test cases were presented, which help to bridge the divide between simple dry test cases and models with fully-coupled sets of physical parametrisations. This included a three-dimensional rising thermal and a moist gravity-wave test which was used to measure the spatial convergence properties of the model, which we found to be at least second-order in both the $k = 0$ and $k = 1$ configurations.

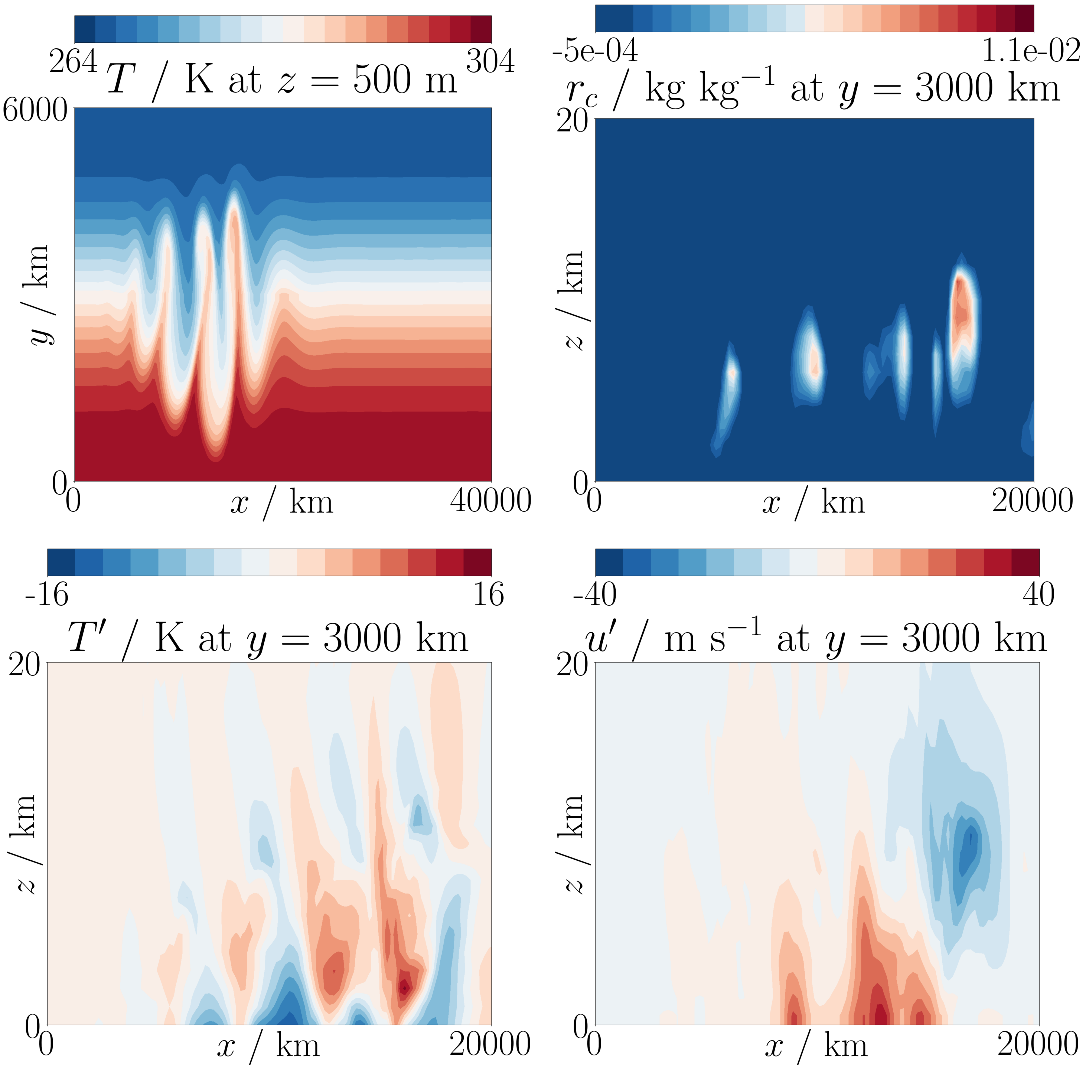


Figure 5.11: Cross-sections of fields at $t = 12$ days from the moist baroclinic wave test case using the $k = 0$ spaces. The grid sizes used were $\Delta x = \Delta y = 200$ km and $\Delta z = 1$ km, with $\Delta t = 300$ s. Shown are (top left) the T field on $z = 500$ m contoured every 2 K, (top right) the r_c field on $y = 3000$ km contoured every 5×10^{-4} kg kg^{-1} with the zero contour omitted, (bottom left) the perturbed temperature field on $y = 3000$ km with contours every 2 K and (bottom right) the perturbed zonal wind u', with contours spaced by 5 m s^{-1}. Values shown on the $y = 3000$ km plane are computed from the lower side of the plane. Taking place in a large-scale three-dimensional channel, this is a more challenging test for the discretisation than the others in Section 5.6. This test more closely simulates those conditions in a real dynamical core used for numerical weather prediction, and in contrast to the test from Section 5.6.4 takes place at much lower resolution. The results do show the formation of the baroclinic wave, with stronger warm and cold fronts than those in the dry case of Figure 4.14.

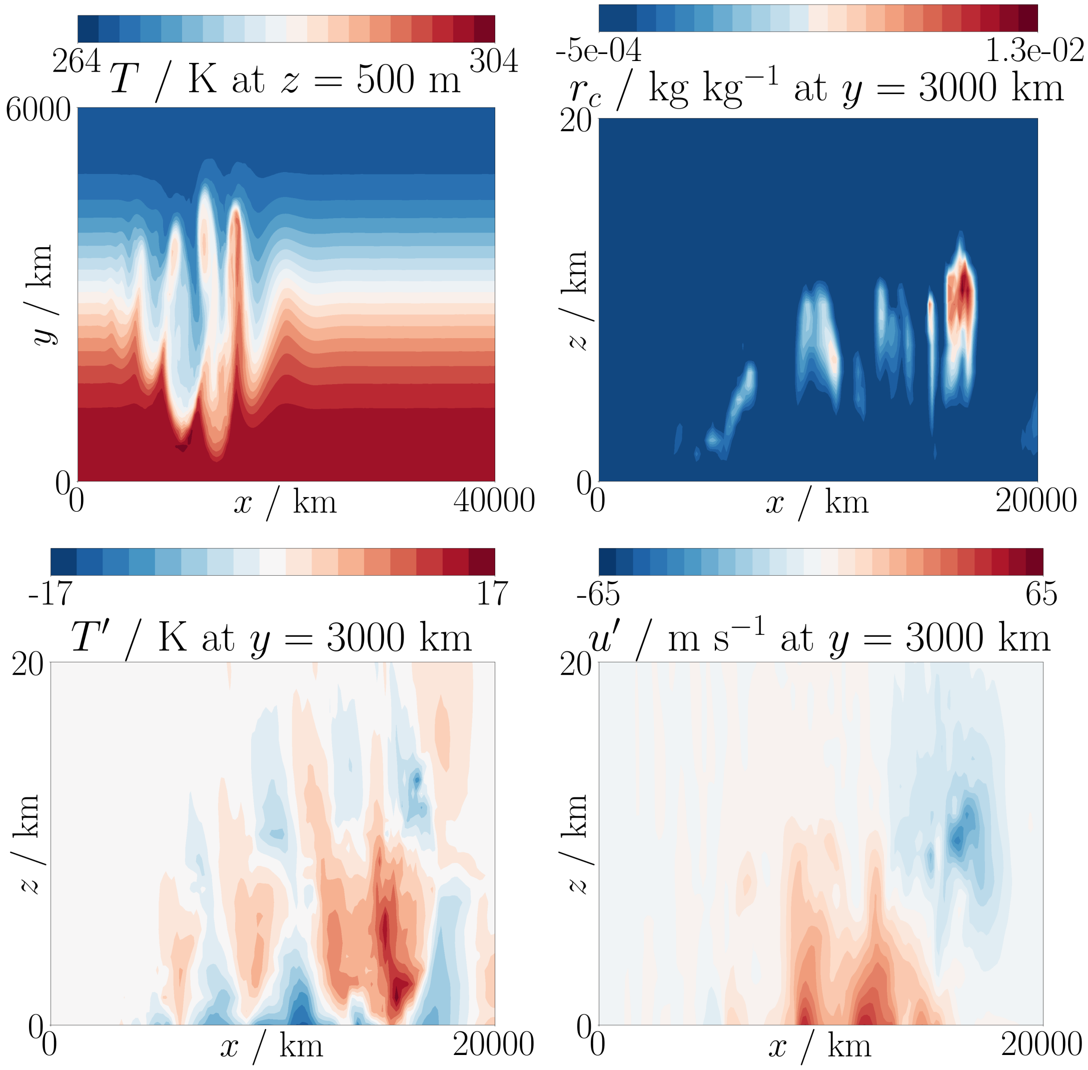


Figure 5.12: Cross-sections of fields at $t = 12$ days from the moist baroclinic wave test case using the $k = 1$ spaces. The grid sizes used were $\Delta x = \Delta y = 250$ km and $\Delta z = 1.5$ km, with $\Delta t = 300$ s. Plots shown are the same as in Figure 5.11, but note the different scale in the zonal wind perturbation plot (bottom right). These results do appear visually different from those presented in Figure 5.11 that used the lowest-order elements. We can see that part of the baroclinic wave has broken, so a less wave-like structure is observed (upper left).

There are two clear strands for future work. In the first, we note that our model involves many choices, such as the choice of family of compatible spaces or the time discretisation used for the microphysics parametrisations. With regards to the coupling of the microphysics routines to one another and the dynamics, the choice of splitting of the routines was arbitrary and should be investigated. To aid further model development, these choices should be explored through a series of controlled tests. One example might be to decouple the microphysics from vertical motions by solving the moist shallow-water equations, such as those of [93] or [94].

The other strand of work involves augmenting the features described in this chapter. For instance, turbulence and radiation parametrisations could be added to the model, or other transport schemes for the lower-order spaces. Simulations over the sphere should be performed, for example that of [95].

II

Stochastic Geophysical Fluids from a Variational Framework

6 Introduction to Stochastic Geophysical Fluids

In this part of the thesis, we explore a framework of stochastic parametrisations, in which the effect of the unresolved processes upon the resolved flow is described through a stochastic contribution to the advecting velocity. In this framework, presented by [11], the noise is introduced through a variational principle so that the circulation theorem still holds. This chapter will outline the key points of the stochastic framework of [11]. Then we look at this stochastic formulation through discretisations of two simple geophysical fluids. Chapter 7 investigates the stochastic quasi-geostrophic equation, looking at the statistical properties of an enstrophy-conserving finite element discretisation of the equation. Finally, the stochastic Camassa-Holm equation and its peakon solutions are explored in Chapter 8.

6.1 Motivation

When attempting to express the effect of unresolved processes upon the resolved flow using parametrisations, the challenge is that the unresolved processes are unknown. In numerical weather prediction models, another large source of uncertainty is in obtaining the initial condition from real observations. Much of the recent skill in weather forecasting over recent decades has come from the advent of data assimilation and ensemble forecasting [96]. Rather than simply running a single model, a modern weather forecast involves an ensemble of simulations with slightly differing initial conditions, each of which evolves along a different forecast trajectory, initially guided by observations of the atmosphere. This provides a way of expressing the forecast uncertainty.

Another way to quantify this uncertainty that has received recent attention is to express the uncertainty in the parametrisations by formulating them with a stochastic component. An introduction to this approach is [97], while a summary of recent developments can be found in [98]. However, as argued by [99], the details of the stochastic approach matter and should be consistent with the physical causes of the uncertainty.

This motivates approaches to stochastic fluid dynamics which preserve some of the physical structure of the deterministic equations. Here we will consider descriptions in which the advection has a stochastic component. Some approaches, such as those of [100] and [101], conserve energy. In contrast, the framework presented by [11] preserves the circulation theorem. This latter formulation is the focus of the remainder of this the-

sis.
We then look to apply the stochastic framework of [11] to simple geophysical fluids. In Chapters 7 and 8, we explore properties of the quasi-geostrophic equation and the Camassa-Holm equation within this framework. The rest of this chapter outlines some background material, briefly describing some concepts in both stochastic calculus and variational approaches to fluid dynamics. Then the formulation of [11] is explained.

6.2 Stochastic Calculus

We begin by considering a process W that is stochastic in time. As a stochastic process, W does not evolve smoothly, meaning it is not differentiable, although it is continuous in time. For this reason, equations containing stochastic processes are typically written containing integrals or differentials rather than derivatives. W can however be written as increments corresponding to time intervals, with dW describing the limit as the time intervals tend to the infinitesimal dt. We choose W to be a *Wiener process*, so that dW is a random variable with independent increments taken from an identical Gaussian probability distribution, which has mean 0 and variance dt. A consequence of this is that

$$(dW)^2 = dt \tag{6.1}$$

and that in terms of a random number $\mathcal{N}(0,1)$, normally distributed with mean 0 and variance 1,

$$dW = \sqrt{dt}\,\mathcal{N}(0,1). \tag{6.2}$$

There are other types of stochastic process, such as Poisson processes or the more general Lévy processes, but here we consider only Wiener processes. These are very common in physical processes due to the *central limit theorem*, which describes how the probability of the sum of independent random variables tends to a Gaussian distribution.

When describing Wiener processes, two types of calculus are natural, each corresponding to different types of physical system. Systems described by Itô calculus consist of discrete jumps which are separated by a time scale that is much smaller than the time scale of interest [99]. Such processes are common in financial mathematics. If the time t is partitioned into discrete intervals, then the Itô integral of a random variable $X(t, W_t)$ can be written as

$$\int_0^t X(t, W_t)\, dW = \lim_{N\to\infty} \sum_{i=0}^{N-1} X(t_i)\left(W_{t_{i+1}} - W_{t_i}\right). \tag{6.3}$$

The expectation of an Itô term in a stochastic differential equation is zero.

In contrast, Stratonovich calculus describes the limit, as the frequency spectrum tends to being uniform, of a process continuous in time with coloured noise [102]. This Stratonovich description occurs more regularly in physics and engineering. The Stratonovich integral, denoted with $\circ$, is

$$\int_0^t X(t, W_t) \circ dW = \lim_{N\to\infty} \sum_{i=0}^{N-1} \tfrac{1}{2} \left(X(t_{i+1}) + X(t_i)\right) \left(W_{t_{i+1}} - W_{t_i}\right). \tag{6.4}$$

Although the expectation of a Stratonovich term is not necessarily 0, unlike its Itô counterpart, Stratonovich calculus is equipped with the usual product rule.

Expansion of (6.3) and (6.4) via a Taylor series, assuming $X(t, W_t)$ is differentiable with respect to both t and W_t, reveals the relationship between the Stratonovich and Itô integrals to be

$$\int_0^t X(t, W_t) \circ dW = \frac{1}{2}\int_0^t \frac{\partial X}{\partial W}\, dt + \int_0^t X(t, W_t)\, dW, \tag{6.5}$$

which shows that transformation between the two calculi introduces a drift term.

Finally, we consider a change of variables using Itô calculus, which is described by *Itô's lemma*. Following [103], we consider the stochastic differential equation

$$dX = f\, dt + g\, dW. \tag{6.6}$$

To derive the equation for the evolution of $Y(X, t)$, it is again expanded as a Taylor series, revealing that

$$dY = \left(\frac{\partial Y}{\partial t} + f\frac{\partial Y}{\partial X} + \frac{g^2}{2}\frac{\partial^2 Y}{\partial X^2}\right) dt + g\frac{\partial Y}{\partial X}\, dW. \tag{6.7}$$

For an informal introduction to stochastic calculus see [103], while a rigorous text is [104].

6.3 Variational Principles for Fluid Dynamics

Just as there are Lagrangian and Hamiltonian descriptions of systems in classical mechanics, the evolution of many fluids can be described with Lagrangian or Hamiltonian formulations. Lagrangian approaches involve using Hamilton's principle: the equations of motion are those that extremise the action S, often subjected to some constraints enforced by Lagrange multipliers. For instance, consider a fluid of velocity $\boldsymbol{v}$ with a density ρ obeying a continuity equation. It satisfies Hamilton's principle for the following action:

$$S = \int_0^t \left[\ell(\boldsymbol{v}, \rho) + \int_\Omega \lambda \left(\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla}\cdot(\rho \boldsymbol{v})\right) dx\right] dt, \tag{6.8}$$

where $\ell(\boldsymbol{v}, \rho)$ is the Lagrangian of the system and λ is the Lagrange multiplier enforcing the continuity equation. Taking free variations, then manipulating, yields the equations of motion of the system, including

$$\frac{\partial}{\partial t}\left(\frac{1}{\rho}\frac{\delta \ell}{\delta \boldsymbol{v}}\right) + \boldsymbol{v}\cdot\boldsymbol{\nabla}\left(\frac{1}{\rho}\frac{\delta \ell}{\delta \boldsymbol{v}}\right) + \frac{1}{\rho}\sum_j \frac{\delta \ell}{\delta v^j}\boldsymbol{\nabla} v^j - \boldsymbol{\nabla}\frac{\delta \ell}{\delta \rho} = 0. \tag{6.9}$$

This approach is commonly known as the Clebsch variational principle, see for instance [105]. Similar derivations can be found in [106] and [107].

Fluids with Hamiltonian structure have equations of motion defined in terms of a Hamiltonian H and a Poisson bracket {F, H}. The time derivative of a variable F of the system is given by

$$\frac{dF}{dt} = \{F, H\}. \tag{6.10}$$

Commonly this is a Lie-Poisson bracket, which depends upon variational derivatives, taking the form

$$\{F, H\}(\boldsymbol{v}) = \left\langle \boldsymbol{v}, \left[\frac{\delta F}{\delta \boldsymbol{v}}, \frac{\delta H}{\delta \boldsymbol{v}}\right]\right\rangle, \tag{6.11}$$

where, for operators A and B, $[A, B] = AB - BA$ is the commutator and $\langle A, B\rangle = \int_\Omega AB\, dx$ is the L^2 pairing.

An elegant language unifying the concepts of variational principles is geometric mechanics. In this description, the evolution of the system is expressed abstractly via a Lie group (a group that is a differentiable manifold) or a group of diffeomorphisms (a differentiable isomorphism between manifolds). In the case of fluids, the smooth evolution of a labelled particle within the fluid is a diffeomorphism.

A central concept within geometric mechanics is the *Lie derivative*, $\mathfrak{L}_{\boldsymbol{v}}$, with respect to some vector field $\boldsymbol{v}$. The definition, from [108], of a Lie derivative $\mathfrak{L}_{\boldsymbol{v}}$ acting upon a differential k-form ω , is given by Cartan's formula:

$$\mathfrak{L}_{\boldsymbol{v}}\omega := \boldsymbol{v} \lrcorner\, d\omega + d(\boldsymbol{v} \lrcorner\, \omega), \tag{6.12}$$

where d is the exterior derivative and $\lrcorner$ represents the interior product, with both described in Chapter 2. This typically plays the role of the advective derivative term, and its action upon a 0-form a and a density $\rho\, dV$ is given by [109] as

$$\mathfrak{L}_{\boldsymbol{v}} a \equiv \boldsymbol{v}\cdot\boldsymbol{\nabla} a, \qquad \mathfrak{L}_{\boldsymbol{v}}\rho\, dV \equiv \boldsymbol{\nabla}\cdot(\rho\boldsymbol{v})\; dV. \tag{6.13}$$

Another entity within geometric mechanics is the *momentum map*, which combines two k-forms q and p, defined by

$$\langle p \diamond q, \boldsymbol{v}\rangle_{\mathfrak{X}} := \langle p, -\mathfrak{L}_{\boldsymbol{v}} q\rangle_V, \tag{6.14}$$

where V is a vector space and $\mathfrak{X}(V)$ is the Lie algebra of vector fields [105]. More information on momentum maps can be found in [110] or [108]. For 0-form a and 3-form $\rho\, dV$, the action of the momentum map is given by

$$p \diamond a = -p\boldsymbol{\nabla} a, \qquad p \diamond \rho\, dV = \rho \boldsymbol{\nabla} p\, dV. \tag{6.15}$$

In the language of geometric mechanics, the analogue of the Euler-Lagrange equation (6.9) is the Euler-Poincaré equation:

$$\frac{\partial}{\partial t}\left(\frac{\delta \ell}{\delta \boldsymbol{v}}\right) + \mathfrak{L}_{\boldsymbol{v}}\left(\frac{\delta \ell}{\delta \boldsymbol{v}}\right) = \frac{\delta \ell}{\delta \rho} \diamond \rho. \tag{6.16}$$

For more on geometric mechanics, see for instance [109].

6.4 The Stochastic Variational Framework

The main result of [11] is the presentation of a stochastic variational framework for fluid dynamics. The advecting field is the stochastic velocity

$$d\boldsymbol{x}_t(\boldsymbol{x}) = \boldsymbol{v}(\boldsymbol{x}, t)\, dt + \sum_i \Xi_i(\boldsymbol{x}) \circ dW_i, \tag{6.17}$$

where the dW_i are Wiener processes governed by the Stratonovich calculus and the $\Xi_i(\boldsymbol{x})$ functions distribute the noise in space. If q is an advected quantity, the stochastically constrained variational principle of [11] is $\delta S = 0$, where the action S is given by

$$S = (\boldsymbol{v}, p, q) = \int \left[\ell(\boldsymbol{v}, q)\, dt + \langle p, dq + \mathfrak{L}_{d\boldsymbol{x}_t} q\rangle\right]. \tag{6.18}$$

This approach was justified by [111], who decomposed the deterministic flow into slow, large-scale motions and fast, small-scale fluctuations. Homogenisation theory was then used to obtain the large-scale flow. The key property of this framework, as shown by [11], is that the circulation theorem is preserved by the stochastic transport. Other examples of investigations of this class of stochastic equations include [112], [113], [114] and [115].

7 On a Discretisation of the Stochastic Quasi-Geostrophic Equation

In this chapter, we apply the stochastic formulation outlined in Chapter 6 to a commonly used simple model within geophysical fluid dynamics: that of the quasi-geostrophic (QG) equation. This builds upon the author's Master's thesis, [116], which presented an enstrophy-conserving finite element discretisation of the stochastic QG equation and investigated its statistical properties. The inspiration for the work in this chapter came from [117], which demonstrated that the statistics of different discretisations of the quasi-geostrophic equation depended upon the properties conserved by the discretisation. We therefore attempt to show that the same statistical theory applies to our discretisation of the stochastic QG equation.

The main new results presented in this chapter are:

- the use of the Metropolis algorithm to generate samples from the statistical distribution, allowing the statistical distribution to be sampled for arbitrary spatial domains, which was not the case for the sampling method used in [116];
- the demonstration of convergence of the statistical distribution to the fluid discretisation, in the limit that the grid spacing goes to zero, thus verifying that the statistical theory applies to our discretisation;
- the examples comparing the discretisation and the statistical distribution over spherical domains, including in the presence of topography. This was not possible with the statistical sampling method used in [116].

This chapter is adapted from our paper [12], where these results first appeared.

7.1 Introduction

The quasi-geostrophic (QG) equation is one of the simplest in the hierarchy of models of geophysical fluid dynamics. It describes a shallow, divergence-free fluid near geostrophic balance – the balance between the Coriolis and pressure gradient forces. The QG equations can be derived by expanding the rotating shallow water equations in terms of a small Rossby number $\mathrm{Ro} \ll 1$, where Ro expresses the size of the inertial forces relative to the rotational forces. The state of the system is defined by the potential vorticity (PV) field q, with the governing equations written as

$$\frac{\partial q}{\partial t} + \boldsymbol{\nabla}^{\perp}\psi \cdot \boldsymbol{\nabla} q = 0, \qquad q := \nabla^2\psi - \mathcal{F}\psi + f, \tag{7.1}$$

where vectors are two-dimensional (with no vertical component), $\mathcal{F}$ is a non-dimensional Froude number, f is the non-dimensional Coriolis parameter and ψ is the stream function related to the velocity $\mathbf{u}$ by

$$\mathbf{u} = \boldsymbol{\nabla}^{\perp}\psi. \tag{7.2}$$

A full derivation and description of the QG system can be found in [118].

In contrast to (7.1), the stochastic QG equation within the framework of [11] is written as

$$dq + d\mathbf{x}_t \cdot \boldsymbol{\nabla} q = 0, \qquad q := \nabla^2\psi - \mathcal{F}\psi + f, \tag{7.3}$$

where $d\mathbf{x}_t$ is given by (6.17) and is the new advecting velocity with a stochastic component. This equation was originally derived in [11] from a Hamiltonian perspective. As in [116] and [12], it can also be obtained by inserting the appropriate Lagrangian into the appropriate analogue of the stochastic variational principle (6.18). This is the approach we use here. Following the framework of [107], we construct the following action

$$\begin{aligned} S[\mathbf{u}, \rho, \phi, \mathbf{p}, \mathbf{q}, P] = \int \ell(\mathbf{u}, \rho, P)\, dt + \int_{\Omega} \phi \left[d\rho + \left(\mathbf{u}\, dt + \sum_i \Xi_i(\mathbf{x}) \circ dW_i \right) \cdot \boldsymbol{\nabla}\rho \right] \\ + \mathbf{p} \cdot \left[d\mathbf{q} + \left(\mathbf{u}\, dt + \sum_i \Xi_i(\mathbf{x}) \circ dW_i \right) \cdot \boldsymbol{\nabla}\mathbf{q} \right] d^2x\, dt, \end{aligned} \tag{7.4}$$

where ℓ will be the Lagrangian, ϕ is the Lagrange multiplier for the continuity equation of ρ, while $\mathbf{p}$ enforces the constraint advection of Lagrangian particles described by $\mathbf{q}(\mathbf{x}, t)$, the back-to-labels map returning the Lagrangian label of the fluid particle at position $\mathbf{x}$ at time t. Here we shall assume that the basis functions Ξ_i are divergence-free, and are tangential to the boundary $\partial\Omega$ of the domain Ω, which we shall assume to be simply connected.

Following [107], we use the specific Lagrangian for QG:

$$\ell(\mathbf{u}, \rho, P) = \int_{\Omega} \left[\frac{1}{2}\rho|\mathbf{u}|^2 - \frac{1}{2}\mathcal{F}\rho\mathbf{u} \cdot \Delta^{-1}\mathbf{u} + \rho\mathbf{u} \cdot \mathbf{R} + P(\rho - \rho_0) \right] d^2x, \tag{7.5}$$

where $\mathbf{R}$ is the fluid velocity due to the rotation of the planet so that $\hat{\mathbf{z}} \cdot \boldsymbol{\nabla} \times \mathbf{R} = f$, and where the operator Δ^{-1} is the inverse of the Laplacian operator.

After computing the Euler-Lagrange equations and eliminating $\mathbf{p}$, $\mathbf{q}$ and ϕ, computations in [11] lead to the equation

$$\mathrm{d}\left(\frac{1}{\rho_0}\frac{\delta\ell}{\delta\mathbf{u}}\right)+\mathrm{d}\boldsymbol{x}_t\cdot\boldsymbol{\nabla}\left(\frac{1}{\rho_0}\frac{\delta\ell}{\delta\mathbf{u}}\right)+\frac{1}{\rho_0}\sum_k\frac{\delta\ell}{\delta u^k}\boldsymbol{\nabla}\mathrm{d}x_t^k+\boldsymbol{\nabla}\frac{\delta\ell}{\delta\rho}\,\mathrm{d}t=0.\quad\frac{\delta\ell}{\delta P}=0. \tag{7.6}$$

For the QG case this gives

$$\frac{1}{\rho_0}\frac{\delta\ell}{\delta\mathbf{u}}=\mathbf{u}+\mathbf{R}-\mathcal{F}\Delta^{-1}\mathbf{u},\quad\frac{\delta\ell}{\delta\rho}=\frac{1}{2}|\mathbf{u}|^2-\frac{1}{2}\mathcal{F}\mathbf{u}\cdot\Delta^{-1}\mathbf{u}+\mathbf{u}\cdot\mathbf{R}+P,\quad\rho=\rho_0. \tag{7.7}$$

Taking the curl of (7.6) and manipulating using vector calculus identities yields

$$(\mathrm{d}+\mathrm{d}\boldsymbol{x}_t\cdot\boldsymbol{\nabla})\left[\hat{\boldsymbol{z}}\cdot\boldsymbol{\nabla}\times\left(\frac{1}{\rho_0}\frac{\delta\ell}{\delta\mathbf{u}}\right)\right]=0. \tag{7.8}$$

For the QG reduced Lagrangian we compute

$$\hat{\boldsymbol{z}}\cdot\boldsymbol{\nabla}\times\left(\frac{1}{\rho_0}\frac{\delta\ell}{\delta\mathbf{u}}\right)=\hat{\boldsymbol{z}}\cdot(\boldsymbol{\nabla}\times\mathbf{u})+\hat{\boldsymbol{z}}\cdot(\boldsymbol{\nabla}\times\mathbf{R})-\mathcal{F}\Delta^{-1}\hat{\boldsymbol{z}}\cdot(\boldsymbol{\nabla}\times\mathbf{u}), \tag{7.9}$$

$$=\nabla^2\psi-\mathcal{F}\psi+f, \tag{7.10}$$

after substituting (7.7) and introducing the stream function ψ so that $\nabla^{\perp}\psi=\mathbf{u}$. The boundary conditions require that $\psi=0$ on $\partial\Omega$. The resulting equation is

$$(\mathrm{d}+\mathrm{d}\boldsymbol{x}_t\cdot\boldsymbol{\nabla})\left[\nabla^2\psi-\mathcal{F}\psi+f\right]=0, \tag{7.11}$$

which is the stochastic QG equation for potential vorticity $q=\nabla^2\psi-\mathcal{F}\psi+f$. This equation has an infinite set of conserved quantities,

$$C^p=\int_\Omega q^p\,\mathrm{d}^2x, \tag{7.12}$$

for $p=1,2,3,\ldots$, with $p=1$ corresponding to the total PV, and $p=2$ proportional to the enstrophy, which is given by $Z=\frac{1}{2}\int_\Omega q^2\,\mathrm{d}^2x$. Although the energy is not conserved, we can deduce that it remains bounded, since

$$2E=\int_\Omega\left(|\nabla\psi|^2+\mathcal{F}\psi^2\right)\,\mathrm{d}^2x, \tag{7.13}$$

$$=\int_\Omega(f-q)\psi\,\mathrm{d}^2x, \tag{7.14}$$

$$\leqslant\left(\int_\Omega(f-q)^2\,\mathrm{d}^2x\right)^{1/2}\left(\int_\Omega\psi^2\,\mathrm{d}^2x\right)^{1/2}, \tag{7.15}$$

$$\leqslant C\left(\int_\Omega q^2\,\mathrm{d}^2x\right)^{1/2}\left(2\int_\Omega|\nabla\psi|^2+\mathcal{F}\psi^2\,\mathrm{d}^2x\right)^{1/2}, \tag{7.16}$$

with C a positive constant having used the Poincaré inequality, and hence

$$E \leqslant \sqrt{C/2}\int_{\Omega} q^2 \, d^2x, \tag{7.17}$$

and so E is bounded by a constant multiplied by the enstrophy, a positive constant of motion.

7.2 Finite Element Discretisation

In this section we describe, for the stochastic QG equation, a discretisation that preserves the total PV and enstrophy. This discretisation was presented in [116] and [12]. We use a finite element discretisation, which allows for the equation to be easily solved on arbitrary meshes, and in particular on the sphere.

The weak form of the stochastic QG equation is obtained by multiplying equation (7.3) by a test function γ, and integrating by parts to obtain

$$d\int_{\Omega} \gamma q \, d^2x - \int_{\Omega} q \boldsymbol{\nabla}\gamma \cdot \left(\boldsymbol{\nabla}^{\perp}\psi \, dt + \sum_i \Xi_i(\boldsymbol{x}) \circ dW_i \right) d^2x = 0, \tag{7.18}$$

where the boundary term vanishes since $d\boldsymbol{x}_t \cdot \hat{\boldsymbol{n}} = 0$ on $\partial\Omega$. A similar procedure, multiplying the relationship between ψ and q by a test function ϕ that vanishes on the boundary leads to

$$\int_{\Omega} \left(\mathcal{F}\phi\psi + \boldsymbol{\nabla}\phi \cdot \boldsymbol{\nabla}\psi \right) d^2x = \int_{\Omega} \phi \left(f - q \right) d^2x. \tag{7.19}$$

This is the standard weak form for the Helmholtz equation.

We introduce a finite element discretisation by choosing a continuous finite element space V, defining

$$\mathring{V} = \{\psi \in V : \psi = 0 \text{ on } \partial\Omega\}. \tag{7.20}$$

The finite element discretisation is obtained by choosing $(q, \psi) \in (V, \mathring{V})$ such that equations (7.18)-(7.19) hold for all test functions $(\gamma, \phi) \in (V, \mathring{V})$.

It follows immediately from the weak form (7.18) that the total PV is conserved, since choosing $\gamma = 1$ leads to

$$d\int_{\Omega} q \, d^2x = 0. \tag{7.21}$$

The enstrophy is conserved since choosing $\gamma = q$ leads to

$$d\int_{\Omega} q^2 \, d^2x = 0. \tag{7.22}$$

Proof that the p-th moment is conserved requires taking $\gamma = q^{p-1}$, but this is not in V for $p > 2$ and thus higher moments are not conserved. In the absence of noise, this discretisation also conserves energy; it reduces to the standard vorticity-stream function finite element formulation. For time integration we use the implicit midpoint rule, and we obtain $\forall \gamma \in V$ and $\forall \phi \in \mathring{V}$,

$$\begin{aligned} \int_\Omega \gamma \left(q^{n+1} - q^n\right) \, d^2x \\ - \int_\Omega \frac{q^{n+1} + q^n}{2} \nabla\gamma \cdot \left(\nabla^\perp \psi^{n+1/2} \Delta t + \sum_i \Xi_i(\mathbf{x}) \Delta W_i \right) d^2x = 0, \end{aligned} \tag{7.23}$$

$$\int_\Omega \left(\mathcal{F}\phi\psi^{n+1/2} + \nabla\phi \cdot \nabla\psi^{n+1/2} \right) d^2x - \int_\Omega \phi \left(f - \frac{q^{n+1} + q^n}{2} \right) d^2x = 0, \tag{7.24}$$

where ΔW_i are independent random variables with normal distribution, $N(0, \Delta t)$. This provides a coupled non-linear system of equations for $(q^{n+1}, \psi^{n+1/2})$ which may be solved using Newton's method.

Taking $\gamma = 1$ immediately gives conservation of the total vorticity Π,

$$\int_\Omega \left(q^{n+1} - q^n\right) \, d^2x = 0. \tag{7.25}$$

All the quadratic invariants of the continuous time equations are conserved by the implicit midpoint rule, meaning this scheme conserves the enstrophy Z exactly as well. The scheme is also unconditionally stable through the use of the implicit midpoint rule.

7.3 Statistical Properties of the Numerical Scheme

One of our main goals in this chapter is to understand the properties of the discretisation presented in section 7.2. We are particularly motivated by the work of [119] and [117], which looked at the statistical mechanics of discretisations of the deterministic equation but with randomised initial states. In particular, [117] looked at how the conservation properties of the discretisation could affect the statistics. In both cases, the distribution of states in phase space was given by a Gibbs distribution. It is therefore of interest whether this same approach could be applied to our discretisation of the stochastic QG equation, and whether this can still be described by the Gibbs distribution.

The statistical mechanics of quasi-geostrophic fluids has previously been studied, for example [120]. More recently, the statistical mechanics of numerical discretisations has been considered. [119] considered Fourier truncations of the QG equation, whilst [117] considered finite difference methods using Arakawa's Jacobian, conserving energy and enstrophy. [117] also considered the other Arakawa schemes that conserve energy but not enstrophy, or conserve enstrophy but not energy, and compared the numerical results with statistics from Gibbs distributions derived under those assumptions. Since the stochastic QG equation in our finite element discretisation does not conserve energy but does conserve enstrophy, we are in exactly this second situation. Following that paper, from the maximum entropy principle with constraints of conserved total vorticity Π and enstrophy Z, we find that the invariant distribution for the finite element discretisation is the Gibbs distribution $\mathcal{G}(\mathbf{Q})$, where $\mathbf{Q}$ is the vector of values describing the discrete q field. For our numerical scheme the probability density function for this Gibbs distribution is

$$\mathcal{G}(\mathbf{Q}) = C^{-1} \exp\left[-\alpha\left(Z(\mathbf{Q}) + \mu\Pi(\mathbf{Q})\right)\right], \tag{7.26}$$

where C, α and μ are parameters providing the constraints of conserved Π, conserved Z and that the integral of the distribution is unity.

In this chapter, we are interested in computing statistics from this distribution and comparing them to what is obtained from time averages over numerical solutions from the finite element discretisation. For example, the expectation of the energy E of the system is then

$$\langle E \rangle = \int_{\mathbb{R}^N} E(\mathbf{Q})\mathcal{G}(\mathbf{Q})\, d\mathbf{Q}. \tag{7.27}$$

In general it is not possible to compute this integral analytically. Our approach is therefore to sample the distribution using a Metropolis algorithm.

Before we do this, we will decompose the state vector $\mathbf{Q}$ into stationary and fluctuating parts:

$$\mathbf{Q} = \langle \mathbf{Q} \rangle + \mathbf{Q}'. \tag{7.28}$$

The components $\{Q_i\}$ of $\mathbf{Q}$ are the coefficients in the finite element discretisation, so that for finite element basis $\{\phi_i\}$ with N components,

$$q = \sum_i^N Q_i \phi_i(x). \tag{7.29}$$

Dubinkina and Frank showed in [117] that the average values $\langle Q_i \rangle$ took a constant value. Following their computation, we evaluate

$$\left\langle \frac{\partial Z}{\partial \mathbf{Q}} + \mu \frac{\partial \Pi}{\partial \mathbf{Q}} \right\rangle_{\mathcal{G}} = \int_{\mathbb{R}^N} \left(\frac{\partial Z}{\partial \mathbf{Q}} + \mu \frac{\partial \Pi}{\partial \mathbf{Q}} \right) C^{-1} \exp\left[-\alpha(Z(\mathbf{Q}) + \mu\Pi(\mathbf{Q}))\right]\, d\mathbf{Q}. \tag{7.30}$$

Inspection of the right hand side reveals that

$$\left\langle \frac{\partial Z}{\partial \mathbf{Q}} + \mu \frac{\partial \Pi}{\partial \mathbf{Q}} \right\rangle = -\alpha^{-1} \int_{\mathbb{R}^N} \frac{\partial}{\partial \mathbf{Q}} \mathcal{G}(\mathbf{Q})\, d\mathbf{Q}, \tag{7.31}$$

and if $\mathcal{G}(\mathbf{Q})$ decays sufficiently fast at infinity then we conclude that

$$\left\langle \frac{\partial Z}{\partial \mathbf{Q}} + \mu \frac{\partial \Pi}{\partial \mathbf{Q}} \right\rangle = 0. \tag{7.32}$$

In the finite element discretisation with domain Ω, $\Pi(\mathbf{Q})$ and $Z(\mathbf{Q})$ are given by

$$\Pi(\mathbf{Q}) = \sum_{i}^{N} \int_{\Omega} \phi_i(\mathbf{x}) Q_i \, d^2x, \quad Z(\mathbf{Q}) = \frac{1}{2} \sum_{i,j}^{N} Q_i Q_j \int_{\Omega} \phi_i(\mathbf{x}) \phi_j(\mathbf{x}) \, d^2x. \tag{7.33}$$

Substituting these into (7.32) gives

$$\left\langle \int_{\Omega} \phi_i(\mathbf{x})(Q(\mathbf{x}) + \mu) \, d^2x \right\rangle = 0, \quad \forall \phi_i. \tag{7.34}$$

This means that $Q(\mathbf{x})$ is the L^2-projection of the constant function $\mu \times 1$ into V, but $\mu \times 1 \in V$, and we conclude that $\langle Q(\mathbf{x}) \rangle = -\mu$ for all i. For an initial value for the fluid simulation of $\Pi(\mathbf{Q}) = \mathcal{P}_0$, this gives $\langle Q_i \rangle = \mathcal{P}_0 / A$, where $A = \int_{\Omega} d^2x$.

The significance of this result is that it is possible to use the Metropolis algorithm to generate samples from the distribution given in equation (7.26) by taking samples from

$$\mathcal{G}'(\mathbf{Q}') = C^{-1} \exp[-Z(\mathbf{Q}')]. \tag{7.35}$$

This generates samples $\mathbf{Q}'$ with $\langle \Pi(\mathbf{Q}') \rangle = 0$ and $\langle Z(\mathbf{Q}') \rangle = \mathcal{Z}'$, which can be transformed into samples of the desired distribution $\mathcal{G}(\mathbf{Q})$ (i.e. with $\langle \Pi(\mathbf{Q}) \rangle = \mathcal{P}_0$ and $\langle Z(\mathbf{Q}) \rangle = \mathcal{Z}_0$) by taking

$$Q_i = \frac{\mathcal{P}_0}{A} + Q_i' \sqrt{\frac{\mathcal{Z}_0}{\mathcal{Z}'} - \frac{\mathcal{P}_0^2}{2A\mathcal{Z}'}}. \tag{7.36}$$

The Metropolis algorithm can therefore be used to take samples from $\mathcal{G}'(\mathbf{Q})$, which avoids the evaluation of the parameter α in (7.26). The Metropolis algorithm finds samples of $\mathcal{G}'(\mathbf{Q}')$ by generating samples from a similar, known distribution, $\mathcal{L}(\mathbf{Q}')$. A given sample $\mathbf{Q}'$ is accepted to be from $\mathcal{G}'(\mathbf{Q}')$ if

$$\frac{\mathcal{G}'(\mathbf{Q}')}{c\mathcal{L}(\mathbf{Q}')} > 1, \tag{7.37}$$

where c is a prescribed constant greater than 1. The Metropolis algorithm also allows us to avoid evaluating the normalisation constants of the distributions. A more detailed description of the Metropolis algorithm can be found for example in [121].

The known distribution $\mathcal{L}(\mathbf{Q}')$ that we use is

$$\mathcal{L}(\mathbf{Q}') = C_{\mathcal{L}}^{-1} \exp\left[-Z_L(\mathbf{Q}')\right]. \tag{7.38}$$

The lumped enstrophy $Z_L(\mathbf{Q}')$ is defined as

$$Z_L(\mathbf{Q}') = \sum_{ij} \int_\Omega \frac{1}{2} Q_i^2 \phi_i(\mathbf{x})\phi_j(\mathbf{x})\, d^2x \equiv \sum_i \frac{1}{2} Q_i^2 M_i^L, \tag{7.39}$$

with lumped mass matrix $M_i^L := \sum_j \int_\Omega \phi_i(\mathbf{x})\phi_j(\mathbf{x})\, d^2x$. This distribution is now straightforward to sample, as

$$\mathcal{L}(\mathbf{Q}') = C_\mathcal{L}^{-1} \exp\left(-\sum_i^N \frac{1}{2} Q_i^2 M_i^L\right) = C_\mathcal{L}^{-1} \prod_i^N \exp\left(-\frac{1}{2} Q_i^2 M_i^L\right). \tag{7.40}$$

The known distribution is then sampled by generating coefficients Q_i' that are normally distributed with mean 0 and variance $1/M_i^L$.

Therefore samples of the Gibbs distribution $\mathcal{G}(\mathbf{Q})$ are found by using the Metropolis algorithm with the known $\mathcal{L}(\mathbf{Q}')$ to get samples of $\mathcal{G}'(\mathbf{Q}')$, and translating the samples $\mathbf{Q}'$ using (7.36). It is important to note that in general a generated sample $\mathbf{Q}$ will not have $\Pi(\mathbf{Q}) = \mathcal{P}_0$ and $Z(\mathbf{Q}) = \mathcal{Z}_0$. Instead the samples will have $\Pi(\mathbf{Q})$ and $Z(\mathbf{Q})$ *distributed* around $\mathcal{P}_0$ and $\mathcal{Z}_0$. As the resolution increases, the distribution of samples will become tighter around the possible states in the discretisation.

7.4 Numerical Results

In this section we compute numerical trajectories of the finite element discretisation, and compare their time averages with statistics computed using the Gibbs-like distribution. The aim was to learn about the properties of the stochastic QG equation using this approach, and to show that the Gibbs-like distribution describes the distribution of the states of discretisation in phase-space, in the limit that the grid spacing goes to zero.

7.4.1 Experimental Set-up

The tests were run on a sphere with unit radius, that is approximated using an icosahedral mesh. The resolution of this mesh can be refined by subdividing the triangular elements at a given resolution into four smaller triangles to obtain the next resolution. The refinement level of the icosahedron is the number of times this process has been repeated to form the mesh. The function space V was P_1. The fluid simulation was run for T time steps of $\Delta t = 1$ and the stochastic part of the stream function was generated using the projections of the first nine spherical harmonics onto the icosahedron. The strength of the stochastic part of the stream function (i.e. the multiplicative constant to the stochastic basis functions) was kept the same for each basis function. It was found not to affect the average values of the simulation, but increasing it did increase the speed with which the averaged values converged to their limits.

For the statistical simulation, the Gibbs-like distribution was used with Π and Z corresponding to the initial condition of the fluid simulation. At the first stage of the statistical simulation, n samples were generated to create an accurate approximation for $\mathcal{Z}'$ to use in equation (7.36). Then the statistical simulation took n samples of the Gibbs-like distribution that were used for the comparison with the statistics of the fluid simulation. Average properties are found for each simulation by taking the mean value of each sample. In the case of the fluid simulation, the system at each time step is considered to be a sample.

7.4.2 Comparison of Mean Fields

As the simulations are run, this generates average states in which q is well-mixed over the domain. Figure 7.1 shows that the average values predicted from the fluid simulation and the Monte Carlo simulation lie very close to one another, even when the fluid simulation starts far from its average state. This figure plots the average Casimirs $\mathcal{C}_3 = \int_\Omega q^3 \, d^2x$ and $\mathcal{C}_4 = \int_\Omega q^4 \, d^2x$ as a function of the number of samples used in calculating that average for a single run of the simulation: we call this the *rolling average* of the simulations and denote it by angular brackets $\langle \cdot \rangle_t$. This plot was from a run at icosahedron refinement level 4, and the fluid simulation was initialised with $q = \sin \lambda$ for latitude λ.

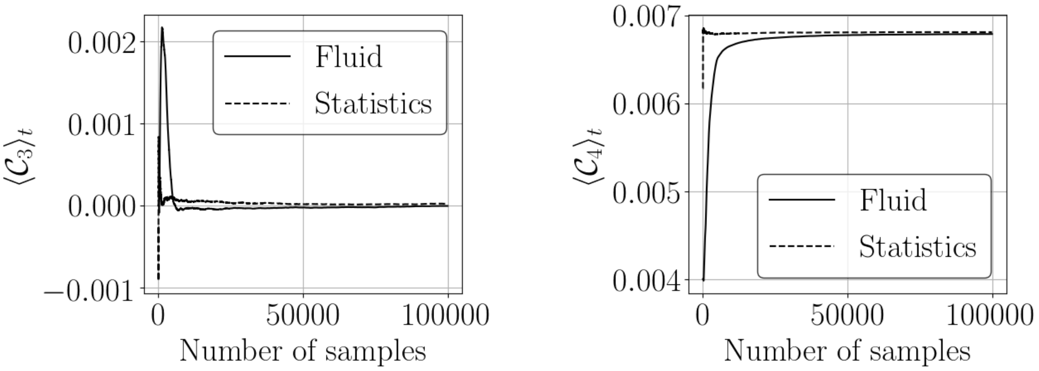


Figure 7.1: The evolution of the rolling average of the Casimirs (left) $\mathcal{C}_3 = \int_\Omega q^3 \, d^2x$ and (right) $\mathcal{C}_4 = \int_\Omega q^4 \, d^2x$ as the stochastic fluid simulation is run for 10^5 time steps and as 10^5 samples are taken from the Gibbs distribution. Both of these are plotted on the same axis, taking one time step of the fluid simulation to be one sample. The fluid simulation was run from an initial condition of $q = \sin \lambda$. One of the key predictions of the statistical theory was that the long-time average values of quantities from the fluid simulation would tend to those averages of samples taken from the statistical distribution (in the limit of the discretisation resolution tending to zero). Here two quantities are shown that are not inherently conserved by the discretisation. These results therefore support the predictions of the statistical theory.

We also took the rolling mean of different diagnostic fields. These were observed to converge to similar fields for both the statistical sampling and from solving the equations of motion. Figure 7.2 shows a comparison of the mean q^2 fields generated via each different method (this is more meaningful than the mean q field, which shows very little variation over the domain). The differences shown over the domain are due to the differences in area of the elements. The fluid simulation results shown in Figure 7.2 (left) are slightly higher than those in 7.2 (right) as at finite resolution the value of Z of the Gibbs distribution is not exactly equal to that of the fluid simulation. As the resolution of the model is refined, the areas of the element will become closer together, and the deviations in the field over the domain should decrease. This was observed and is plotted in Figure 7.3.

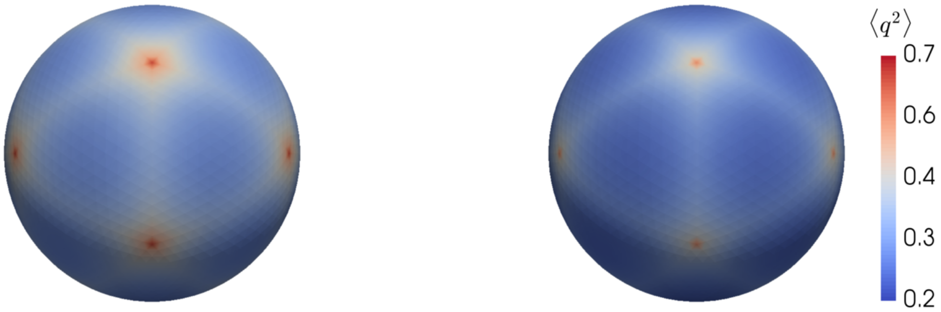


Figure 7.2: A comparison of the mean square potential vorticity q^2 field generated at icosahedron refinement level 4 by (left) the stochastic fluid simulation run for 10^5 steps with an initially random field and (right) 10^5 samples taken from the Metropolis algorithm. In contrast to figure 7.1, this shows average field values rather than average global values. We observe that the two fields are essentially the same, as predicted by the statistical theory. These fields are not uniform over the domain as the element size is not uniform over the domain.

7.4.3 The Effect of $\mathcal{F}$

The effect of the constant $\mathcal{F}$ upon the convergence of the model was also investigated. The equations of motion were solved for a series of different values for $\mathcal{F}$. For each value of $\mathcal{F}$, the fluid model was run 50 times, creating an ensemble with different realisations of the noise. Our aim is to learn about the behaviour of the discretised stochastic QG equation by initialising the fluid at a state far from being well-mixed, and looking at the effect $\mathcal{F}$ has on the rate of mixing under stochastic noise. The initial condition was chosen to be $q = \sin\lambda$, which is a state far from a well-mixed equilibrium.

The difference between the initial value of $\mathcal{C}_4$ and its value for the ensemble average was recorded after $T = 2000$ time steps. We found that the average change in $\mathcal{C}_4$ over this time was a good proxy for the rate of mixing of the PV field: the larger the difference, the

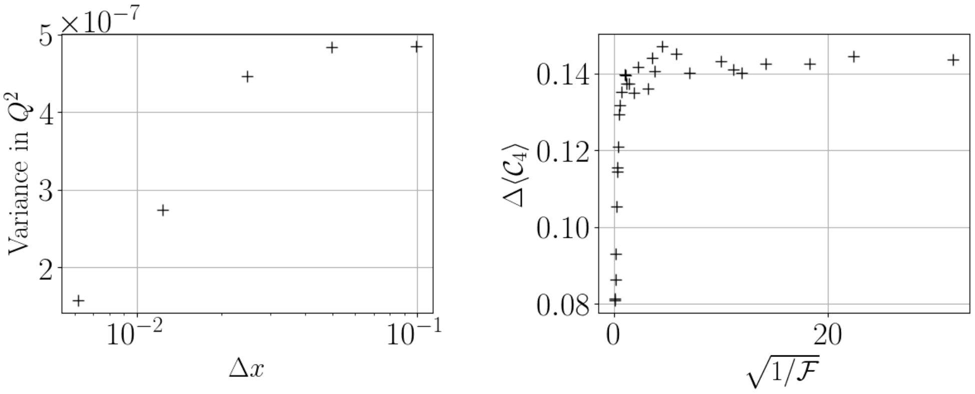


Figure 7.3: (Left) a plot of the final value of $\int_\Omega (q^2 - \bar{q}^2)^2 \, d^2x$ after 10^4 time steps at $\Delta t = 1$ of the fluid simulation, where $\bar{q^2} = \int_\Omega q^2 \, d^2x/A$ and $A = \int_\Omega d^2x$. This shows that as the resolution is increased, the variations in the mean q^2 field reduce. (Right) the rate of mixing (i.e. rate of convergence of statistics to the equilibrium values) of the fluid simulation as a function of the parameter $\mathcal{F}$, as measured by the speed at which the fourth Casimir $\mathcal{C}_4$ moves away from its initial value. Each point represents the difference at the 2000th time step of the ensemble mean value of $\mathcal{C}_4$ from its initial value. We observe that the rate of mixing is low for large $\mathcal{F}$, which corresponds to large (dimensionless) Rossby deformation radii. This is because scales that are below the Rossby deformation radius are effectively just transported by the flow field without feeding back, and so a large $\mathcal{F}$ places more scales in this category.

higher rate of mixing. This was plotted as a function of $1/\sqrt{\mathcal{F}}$, which describes a characteristic length scale. These values are displayed in Figure 7.3, which shows smaller differences for smaller length scales (or higher values of $\mathcal{F}$). This experiment was also performed at different resolutions, which showed the same behaviour.

7.4.4 Convergence with Resolution

Solving the equations of motion produces only samples with identical Π and Z. In contrast, the samples taken from the Gibbs distribution will have differing values of Π and Z, but spread about those values specified in the distribution. As the resolution is increased, samples taken from the Gibbs distribution should fit more tightly around the samples taken from solving the equations of motion.

To investigate this we ran the fluid simulation and sampled the Gibbs distribution at various resolutions to produce histograms of various statistics. We took 10000 samples from the Gibbs distribution without scaling the resultant $\mathbf{Q}$ field (i.e. the samples had $\langle \Pi(\mathbf{Q}) \rangle = 0$ and $\langle Z(\mathbf{Q}) \rangle = Z'$ as described in section 7.3, where Z' will be resolution dependent). This corresponds to a fluid simulation with an initial state of $\Pi = 0$ and $Z = Z'$. Histograms at different resolutions of the Π and Z values of the statistical samples are plotted in Figure 7.4. They have been normalised to remove the resolution dependence of Z'.

The resultant histograms do indeed fit more tightly around the conserved fluid values as the resolution is increased. Figure 7.5 plots the standard deviation of these histograms as a function of resolution, showing linear convergence. We also would expect to see similar behaviour for other statistics, although in the case of more complicated statistics such as variances or higher moments of the distribution, we expect that a very large number of samples will be required in order to observe this convergence.

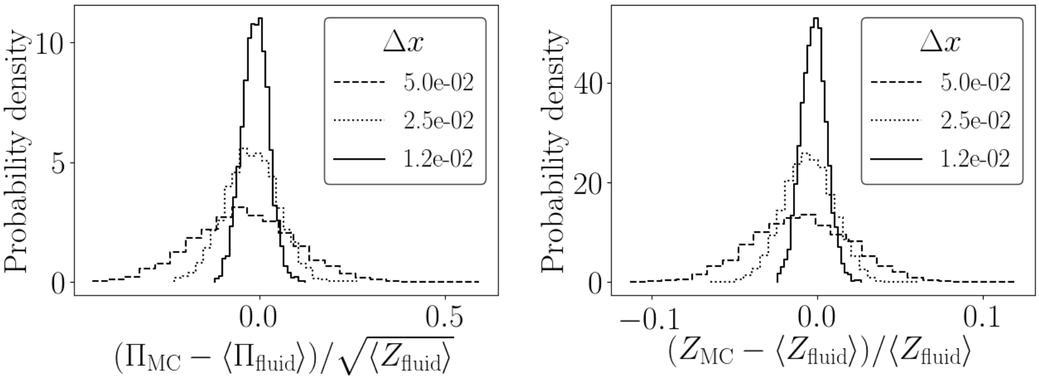


Figure 7.4: Histograms of the (left) $\Pi = \int_\Omega q \, d^2x$ and (right) $Z = \frac{1}{2}\int_\Omega q^2 \, d^2x$ values of different samples of the Gibbs distribution at three resolutions. The statistical theory predicted that the Gibbs distribution should converge to the probability distribution describing the discretisation as the grid spacing tends to zero. These histograms represent this, showing that global values of samples from the Gibbs distribution fit more tightly around that from the fluid discretisation as the grid spacing is reduced.

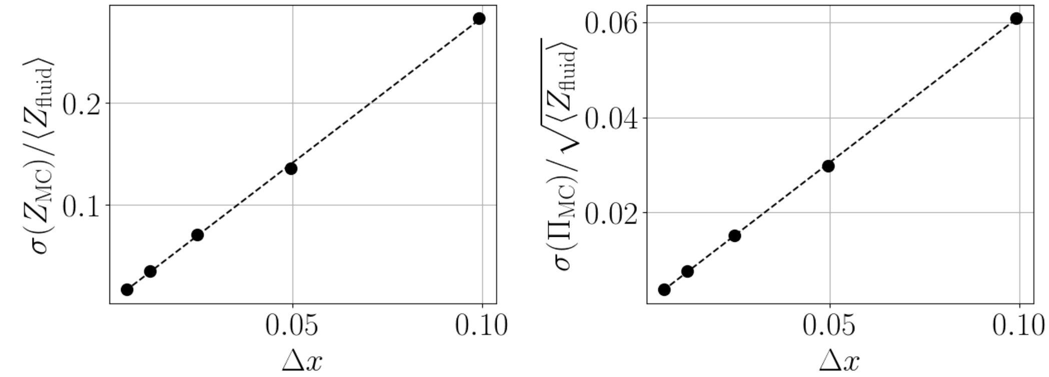


Figure 7.5: The standard deviations of the histograms such as those in Figure 7.4 of (left) $\Pi = \int_\Omega q \, d^2x$ and (right) $Z = \frac{1}{2}\int_\Omega q^2 \, d^2x$ as a function of resolution. This shows the convergence of the Gibbs distribution to the delta function as the grid spacing goes to zero, thus supporting the prediction that the fluid discretisation is described by the Gibbs distribution as the grid spacing is reduced to zero.

7.4.5 Topography

The fluid and statistical models also show the same properties when topography is included in the model. This is done by the addition of an extra term to the potential vorticity definition:

$$q := \nabla^2\psi - \mathcal{F}\psi + f + h. \tag{7.41}$$

We performed similar experiments to those described above by mimicking an isolated mountain, as described as the fifth test case in [76]. In this case, the topography is described by the following function:

$$h = h_0(1 - r/R), \tag{7.42}$$

where $h_0 = 2$, $R = \pi/9$ and $r = \min\left[R^2, (\lambda - \lambda_c)^2 + (\theta - \theta_c)^2\right]$, for latitude λ and longitude θ. The centre of the mountain is at $\lambda_c = 3\pi/2$ and $\theta_c = \pi/6$. Figure 7.6 shows comparisons between the fluid and statistical simulations of mean ψ fields plotted after $T = 10^5$ time steps and $n = 10^5$ samples were taken. The third icosahedron refinement level was used. For both simulations, the average fields are very similar.

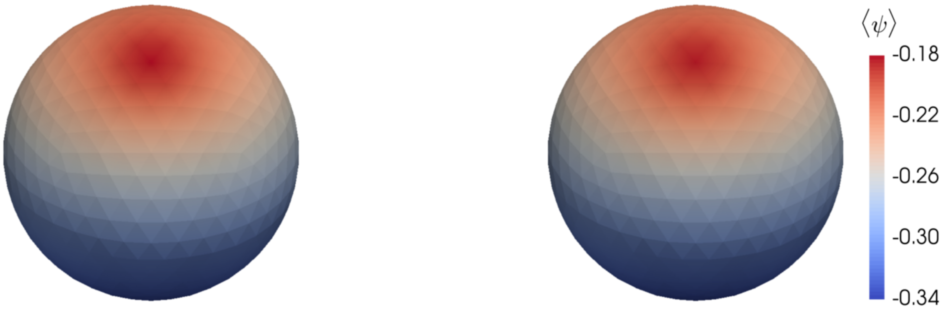


Figure 7.6: A comparison of the mean stream function ψ field generated over topography. In this case we used flow over a single mountain, and we show (left) the stochastic fluid simulation run for 10^6 steps and (right) 10^6 samples taken from the Metropolis algorithm. The plots are essentially the same, illustrating that the predictions of the statistical theory also hold when topography is included in the model.

7.5 Summary and Outlook

The work of [11] gave the framework for a stochastic variational principle for QG, with this equation derived from a Lagrangian perspective in [116]. [116] then presented a finite element discretisation for the stochastic QG equation that conserved the enstrophy of the system. Following [117], which showed that different discretisations of the deterministic QG equation had different statistical properties, we looked at the statistics of the discretisation for the stochastic QG equation. Statistical predictions were then made about this discretisation following the approach of [119], which were tested by sampling using a Metropolis algorithm (in contrast to [116]). The statistics generated by sampling the resulting Gibbs-like distribution were compared with those averaged quantities from solving the stochastic equations of motion, showing that the Gibbs-like distribution does indeed describe the enstrophy-conserving discretisation of the stochastic QG equation.

The stochastic variational framework of [11] offers a promising approach to stochastic parametrisation of the effects of unresolved processes upon the resolved flow. It provides a particularly elegant way of coupling these processes together whilst preserving the circulation theorem of the fluid. Two possible applications are describing the backscatter of unresolved motions on the resolved scales and the use of ensemble forecasting with data assimilation, in which the nearby trajectories of the state can preserve physical properties of the system. However, as demonstrated in this chapter, the statistical properties of such a stochastic approach within a numerical discretisation depend upon the properties of the discretisation itself. This is an important concept to bear in mind when designing a stochastic parametrisation, which is used to capture the statistics of the underlying flow.

8 Peakons in the Stochastic Camassa-Holm Equation

In this final chapter, we consider the application of the stochastic framework of [11] to a shallow-water wave system. We look at the stochastic Camassa-Holm equation, which has been already been studied by [115] and [112]. We develop the work of [112], which investigated the formation of peaked soliton solutions – known as *peakons* – in the stochastic Camassa-Holm equation. In the deterministic case, a smooth, positively-valued initial condition with a negative slope and an inflection point always forms a peakon in finite time [122]. [112] found that peakons form under the same conditions in the stochastic case but only with non-zero probability (and not necessarily a probability of one). Building on this work, we numerically investigate the formation of such peakons in the presence of the stochastic approach to parametrisation of [11].

The results of this chapter are:

- demonstration that the stochastic Camassa-Holm admits weak peakon solutions when the equation is written in hydrodynamic form, which was not explicitly shown by [115] or [112];
- presentation of a finite element discretisation to the hydrodynamic form of the stochastic Camassa-Holm equation, which is a natural way to discretise the equation, with peakons being weak solutions of the stochastic Camassa-Holm equation;
- demonstration that when describing peakons, the finite element discretisation of the stochastic Camassa-Holm equation converges to the (ordinary) stochastic differential equations for the evolution of the peakons, which is important for using the discretisation to investigate the behaviour of peakons;
- numerical experiments supporting the argument that peakons always form in the stochastic Camassa-Holm equation, providing evidence for the unanswered question of [112].

This chapter is adapted from our paper [15], where these results are also presented.

8.1 Background

8.1.1 The Deterministic Camassa-Holm Equation

In this chapter, we build on the work of [115] and [112] in exploring the properties of the Camassa-Holm equation within the stochastic framework of [11]. The Camassa-Holm equation was introduced in [122], and describes one-dimensional dispersive shallow-water waves. It has a bi-Hamiltonian structure and possesses special soliton solutions, which propagate at constant velocity without changing shape or amplitude, as non-linear and dispersive effects cancel out. Here we briefly present the deterministic equation in three different forms.

The first form of the Camassa-Holm equation, for velocity u is

$$u_t - \alpha^2 u_{xxt} + 3uu_x = 2\alpha^2 u_x u_{xx} + \alpha^2 u u_{xxx}, \tag{8.1}$$

with subscripts x and t denoting partial derivatives with respect to spatial coordinate x and time t respectively, and α as a parameter.

The Hamiltonian structure of (8.1) is commonly exploited to write it as the coupled equations

$$m = u - \alpha^2 u_{xx}, \tag{8.2a}$$

$$m_t + (mu)_x + mu_x = 0, \tag{8.2b}$$

where m can be treated as a momentum that is related to u via a Helmholtz operator. The velocity u can be found via the inverse of the Helmholtz operator, which we denote by

$$u = K * m. \tag{8.3}$$

A third form of (8.1), which we call the hydrodynamic form, can be derived by first writing (8.2b) as an operation upon m, which combined with the definition of m gives the following equation for u:

$$\left(\partial_t + \partial_x u + u_x\right)\left(1 - \alpha^2 \partial_x^2\right) u = 0. \tag{8.4}$$

The left hand side describes the action of two operators, and we find the result of alternating their order by introducing the commutator, so that

$$\left(1 - \alpha^2 \partial_x^2\right)\left(\partial_t + \partial_x u + u_x\right) u = -\left[\partial_t + \partial_x u + u_x, 1 - \alpha^2 \partial_x^2\right] u. \tag{8.5}$$

Inspection of the commutator shows that it can be written as the gradient of a potential. Applying the inverse Helmholtz operator to both sides, which commutes through the gradient, gives

$$\left(\partial_t + \partial_x u + u_x\right) u = -\partial_x K * \left[\alpha^2 \left(u^2\right)_{xx} + \frac{\alpha^2}{2}\left(u_x\right)^2\right] \tag{8.6}$$

Noticing that the non-linear term on the left can also be written as a gradient, and using $K * (u - \alpha^2 u_{xx}) = u$, gives the final hydrodynamic form of the Camassa-Holm equation,

$$u_t = -\partial_x \left\{\frac{1}{2}u^2 + K * \left[u^2 + \frac{\alpha^2}{2}\left(u_x\right)^2\right]\right\}. \tag{8.7}$$

The significance of this form of the equation is that no second derivative of u directly appears in the equation. This equation admits weak peakon solutions, as will be discussed in Section 8.2.

Equations (8.1), (8.2) and (8.7) are equivalent ways of expressing the Camassa-Holm equation.

8.1.2 The Stochastic Camassa-Holm Equation

The deterministic equation (8.2) can be found by exploiting the Hamiltonian structure:

$$m_t = -\left(\partial_x m + m\partial_x\right)\frac{\delta H}{\delta m}, \tag{8.8}$$

with the Hamiltonian

$$H = \frac{1}{2}\int_\Omega \left(u^2 + \alpha^2 u_x^2\right)\, dx = \frac{1}{2}\int_\Omega m\left(K * m\right)\, dx, \tag{8.9}$$

in which case $\delta H/\delta m = u$. A similar approach can be used to find the stochastic Camassa-Holm equation, as used in [112]. In this case, we introduce the stochastic Hamiltonian

$$d\mathcal{H} = \frac{1}{2}\int_\Omega m\left(K * m\right)\, dt\, dx + \int_\Omega m \sum_j \Xi^j \circ dW_j\, dx. \tag{8.10}$$

The respective variational derivative is then $\delta d\mathcal{H}/\delta m = dx_t$, with dx_t from (6.17). The resulting stochastic equation is then

$$dm = -\left(m\, dx_t\right)_x - m\left(dx_t\right)_x, \tag{8.11}$$

which can be thought of as the stochastic case of (8.2). This equation also possesses a hydrodynamic form, which can also be obtained by considering the commutator

$$\begin{aligned}&\left[d + \partial_x\, dx_t + \partial_x\left(dx_t\right), 1 - \alpha^2\partial_x^2\right] u = \\ &\qquad \alpha^2\left(2u_{xx}\left(dx_t\right)_x + 5u_x\left(dx_t\right)_{xx} + 2u\left(dx_t\right)_{xxx}\right).\end{aligned} \tag{8.12}$$

Separating this into deterministic and stochastic parts gives

$$
\begin{aligned}
(\mathrm{d} + \partial_x \mathrm{d}x_t + (\mathrm{d}x_t)_x)\, u = &-K * \left[\alpha^2 \left(7u_x u_{xx} + 2uu_{xx}\right)\right] \mathrm{d}t \\
&- \sum_j K * \left[\alpha^2 \left(2u_{xx}\Xi^j_x + 5u_x\Xi^j_{xx} + 2u\Xi^j_{xxx}\right)\right] \circ \mathrm{d}W_j,
\end{aligned}
\tag{8.13}
$$

which results in

$$
\begin{aligned}
\mathrm{d}u = &- \partial_x \left\{\frac{1}{2}u^2 + K * \left[u^2 + \frac{\alpha^2}{2}\left(u_x\right)^2\right]\right\} \mathrm{d}t \\
&- \sum_j \left\{u_x\Xi^j + K * \left[2u\Xi^j_x + \alpha^2 u_x \Xi^j_{xx}\right]\right\} \circ \mathrm{d}W_j.
\end{aligned}
\tag{8.14}
$$

Unlike (8.7), the right hand side of (8.14) cannot be written purely as a gradient, but it also contains no double derivatives of u.

In this chapter, we extend the work of [115], which originally derived the stochastic Camassa-Holm equation (8.11). The authors then presented a discretisation of the stochastic ordinary differential equations describing the evolution of peakons, and used this discretisation to explore the interaction of peakons. The other main work which we develop is that of [112], who showed the probability of peakon formation in this regime is positive, though not necessarily unity. This can compared with the deterministic case, in which peakons always form from an inflection point upon a negative slope within finite time. We attempt to extend this by investigating peakon formation numerically.

The structure of the remainder of this chapter is: in Section 8.2, we verify that peakons do indeed satisfy the stochastic Camassa-Holm equation by writing it in hydrodynamic form. In Section 8.3 we present a finite element discretisation for the stochastic Camassa-Holm equation, showing its numerical convergence properties in Section 8.4. Finally, we numerically investigate the steepening lemma of [112] using our discretisation in Section 8.5.

8.2 Peakon Solutions to the Stochastic Camassa-Holm Equation

The equation that we are attempting to solve

$$
\begin{aligned}
\mathrm{d}u = &- \partial_x \left\{\frac{1}{2}u^2 + K * \left[u^2 + \frac{\alpha^2}{2}\left(u_x\right)^2\right]\right\} \mathrm{d}t \\
&- \sum_j \left\{u_x\Xi^j + K * \left[2u\Xi^j_x + \alpha^2 u_x \Xi^j_{xx}\right]\right\} \circ \mathrm{d}W_j.
\end{aligned}
\tag{8.15}
$$

This admits weak peakon solutions, of the form

$$u = K * [\delta(x - q(t))\, p(t)] = p(t) e^{-|x-q(t)|/\alpha}, \tag{8.16}$$

in which $q(t)$ represents the position of the peakon and $p(t)$ represents its momentum, and together they form a pair of canonical coordinates satisfying a Hamiltonian system. As stated in [112], the Hamiltonian of this stochastic system of M peakons can be written as

$$dh = \frac{1}{4} \sum_{a,b=1}^{M} p_a(t) p_b(t) e^{-|q_a(t)-q_b(t)|/\alpha} \tag{8.17}$$

The evolution of these canonical coordinates is described by the pair of coupled stochastic differential equations

$$dp_a = -\frac{\partial dh}{\partial q_a} = -p_a(t) \frac{\partial dx_t(q_a(t))}{\partial q}, \tag{8.18a}$$

$$dq_a = \frac{\partial dh}{\partial p_a} = dx_t(q_a(t)), \tag{8.18b}$$

where

$$dx_t(q_a(t)) = u(q_a(t))\, dt + \sum_i \Xi^i(q_a(t)) \circ dW^i. \tag{8.19}$$

Considering a single peakon and writing $u = p(t)\exp(-|x - q(t)|)$, gives

$$dx_t(q(t)) = p(t)\, dt + \sum_j \Xi^j(q(t)) \circ dW^j. \tag{8.20}$$

In [112] it was argued that substitution of (8.18) into (8.16) yields (8.11). However, the peakon solution (8.16) only makes sense as a weak solution to (8.14), the verification of which is the focus of this section. In other words, they are solutions to

$$\begin{aligned} \int_\Omega \phi du\, dx = & \int_\Omega \phi_x \left\{ \frac{1}{2} u^2 + K * \left[u^2 + \frac{\alpha^2}{2} (u_x)^2 \right] \right\} dt\, dx \\ & - \sum_j \int_\Omega \phi \left\{ u_x \Xi^j + K * \left[2u\Xi^j_x + \alpha^2 u_x \Xi^j_{xx} \right] \right\} dx \circ dW_j, \end{aligned} \tag{8.21}$$

for all $\phi \in H^1$.

To verify that such peakons do indeed satisfy this equation, we take

$$\phi = A\cos(kx) + B\sin(kx), \qquad \Xi^j = C_j \cos(jx) + D_j \sin(jx). \tag{8.22}$$

Using (8.16) and (8.18), we write the differential $\mathrm{d}u$ as

$$\begin{aligned} \mathrm{d}u &= \frac{u}{p}\mathrm{d}p - \frac{u}{\alpha}\mathrm{sign}(x-q)\mathrm{d}q, && (8.23) \\ &= -\frac{pu}{\alpha}\mathrm{sign}(x-q)\,\mathrm{d}t - u\sum_j \left[\frac{1}{\alpha}\Xi^j(q)\mathrm{sign}(x-q) + \Xi^j_x(q)\right] \circ \mathrm{d}W_j. && (8.24) \end{aligned}$$

The $K*$ operation is performed upon some $f(x)$ as

$$K * f = \int_\Omega e^{-|x-y|/\alpha} f(y)\,\mathrm{d}y. \tag{8.25}$$

Taking the domain Ω as $\mathbb{R}$ and combining these elements to compute the integrals, both right and left hand sides of (8.21) return

$$\begin{aligned} &\frac{2p^2\alpha k}{1+\alpha^2k^2}\left[B\cos(kq) - A\sin(kq)\right]\mathrm{d}t \\ &- \frac{2p\alpha}{1+\alpha^2k^2}\sum_j \big[(BC_jk - AD_jj)\cos(jq)\cos(kq) - (AC_jk + BD_jj)\cos(jq)\sin(kq) \\ &+ (AC_jj + BD_jk)\sin(jq)\cos(kq) + (BC_jj - AD_jk)\sin(jq)\sin(kq)\big] \circ \mathrm{d}W_j, \end{aligned} \tag{8.26}$$

thus verifying that $u = p(t)\exp\left[-|x-q(t)|/\alpha\right]$ is a solution to (8.21), if p and q satisfy (8.18).

8.3 Finite Element Discretisation

Here we present a mixed finite element discretisation of the stochastic Camassa-Holm equation in the case of $c_0 = 0$ and $\gamma = 0$, inspired by equation (8.15), to have confidence in the discrete representation of peakons.

Considering the space V of continuous linear functions, we seek to find the

$$\left(u^{(n+1)}, \Delta F^{(h)}, \Delta G^{(h)}\right) \in (V, V, V) \tag{8.27}$$

that satisfy, for all $(\psi, \phi, \zeta) \in (V, V, V)$, the equations

$$\int_\Omega \left[\psi\left(u^{(n+1)} - u^{(n)}\right) + \psi u^{(h)}_x \Delta v^{(h)}\,\mathrm{d}x - \psi_x \Delta F^{(h)} + \psi \Delta G^{(h)}\right]\,\mathrm{d}x = 0, \tag{8.28a}$$

$$\int_\Omega \left[\phi\Delta F^{(h)} + \alpha^2\phi_x\Delta F^{(h)}_x - \phi u^{(h)}u^{(h)}\Delta t - \frac{\alpha^2}{2}u^{(h)}_x u^{(h)}_x \Delta t\right]\,\mathrm{d}x = 0, \tag{8.28b}$$

$$\int_\Omega \left[\zeta\Delta G^{(h)} + \alpha^2\zeta_x\Delta G^{(h)}_x - \sum_j\left(\int_\Omega 2\zeta u^{(h)}\Xi^j_x - \alpha^2 u^{(h)}\Xi^j_{xx}\right)\Delta W_j\right]\,\mathrm{d}x = 0, \tag{8.28c}$$

where Δt is the time step, $u^{(n)}$ is the value of u at the n-th time level, and we have used the implicit midpoint rule to discretise in time, so that $u^{(h)} = \frac{1}{2}(u^{(n+1)} + u^{(n)})$. The stochastic velocity $\Delta v^{(n+1/2)}$ is given by

$$\Delta v^{(n+1/2)} = u^{(n+1/2)}\Delta t + \sum_j \Xi^j \Delta W^j, \tag{8.29}$$

and ΔW^j is a normally distributed number with zero mean and variance Δt.

8.4 Convergence of the Discretisation to Peakon Solutions

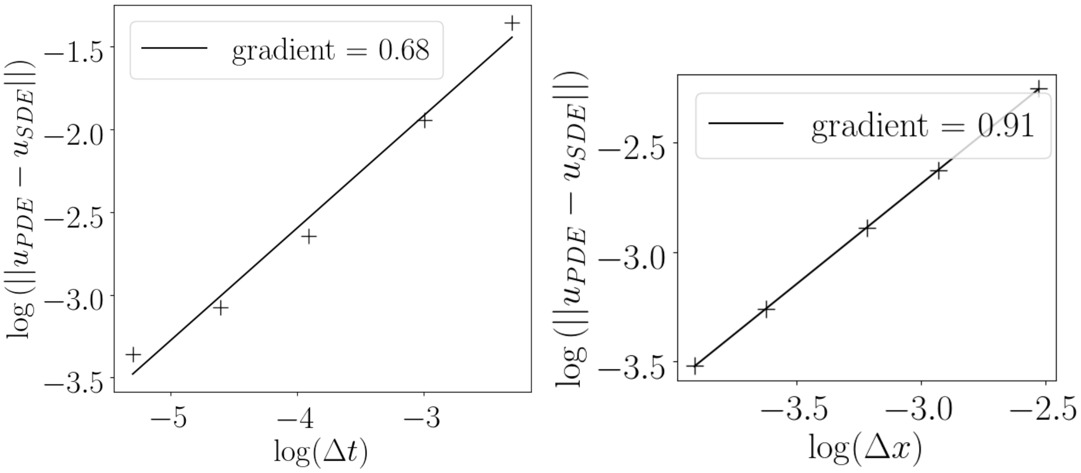


Figure 8.1: (Left) a plot showing the strong convergence of the numerical discretisation with Δt. This plot thus shows that as $\Delta t \to 0$, the solution converges for individual realisations of the noise. (Right) a plot showing the convergence as $\Delta x \to 0$ of the discretisation from Section 8.3 of the stochastic Camassa-Holm equation to the equations (8.30) describing the evolution of peakon solutions. For both plots, the true solution was computed by solving (8.30), and using this to create the u field corresponding to a peakon solution of momentum p and position q within the discretisation from Section 8.3. Crosses mark the data points, with a solid best fit line overlaid.

In this section, we show that when describing the evolution of a peakon, our discretisation presented in Section 8.3 converges to the equations

$$dp = \frac{p}{2}\sum_j \left[\left(\Xi^j_x(q)\right)^2 - \Xi^j(q)\Xi^j_{xx}(q)\right] dt - p\sum_j \Xi^j_x(q)dW_j, \tag{8.30a}$$

$$dq = \left[p + \frac{1}{2}\sum_j \Xi^j(q)\Xi^j_x(q)\right] dt + \sum_j \Xi^j(q)\, dW_j. \tag{8.30b}$$

To do this, we set up a periodic one-dimensional domain of length $L_d = 40$, and taking $\alpha = 1$, we specify an initial condition of

$$u = \begin{cases} \exp\left[(x - L_d/2)/\alpha\right], & x < L_d/2, \\ \exp\left[-(x - L_d/2)/\alpha\right], & x \geqslant L_d/2. \end{cases} \tag{8.31}$$

First we establish a numerical solution of (8.30) that is well-resolved temporally, by solving (8.30) using a simple forward Euler timestepping scheme, with initial condition of $p = 1$ and $q = 20$ and with $\Delta t = 10^{-5}$. For the stochastic basis functions, we used only a single function:

$$\Xi^0 = 0.02 \sin\left(\frac{4\pi x}{L_d}\right). \tag{8.32}$$

Then we performed two tests, one showing the strong convergence of the discretisation (8.28) to this high resolution solution of (8.30) as Δt approached 10^{-5}. This was performed at fixed spatial resolution with $\Delta x = 40/2000$. The second test showed convergence of (8.28) to (8.30) as $\Delta x \to 0$, using $\Delta t = 10^{-5}$.

When solving our discretisation (8.28) with a larger time step, say $n\Delta t$, to ensure that the realisation of the noise corresponded to those with Δt, the random number used for that larger time step would be $\frac{1}{\sqrt{n}}\sum_i^n \Delta W$, where ΔW are the random numbers corresponding to the smaller Δt. At a given time step, the p and q calculated from (8.30) were translated into the u field corresponding to a peakon in the discretisation of the PDE using (8.16), which could be compared with the u found from solving (8.28).

Results showing the strong temporal convergence of the discretisation are shown in Figure 8.1 (left), whilst Figure 8.1 (right) displays the results of the spatial convergence of the discretisation to the underlying stochastic differential equations for the evolution of the peakon.

8.5 Numerical Investigations of the Steepening Lemma

One of the key results of [112] was the investigation of the steepening lemma for the stochastic Camassa-Holm system. In the deterministic case, a peakon will always form in finite time from an inflection point on a negative slope, which was one of the key results of [122]. In [112], it was shown that there was a non-zero probability of a peakon forming in these conditions in the stochastic Camassa-Holm system. In this section we attempt to investigate, numerically, whether this probability is simply non-zero or almost unity.

To add to this discussion, we perform numerical simulations of the stochastic Camassa-Holm equation, using the discretisation of Section (8.3), under many different realisations of the noise, and record the formation of peakon solutions. The first step in this investigation is to determine a numerical diagnostic for whether a solution is in fact a peakon. Our diagnostic used is the distance between inflection points in u in the discretisation, which we denote as μ. For a smooth peak, as $\Delta x \to 0$, μ will converge to some non-zero value, whilst for a peaked solution, μ will converge to zero as $\Delta x \to 0$.

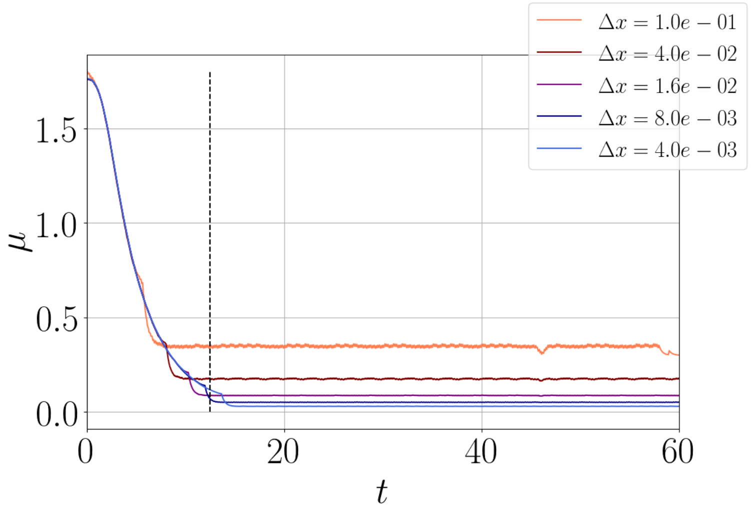


Figure 8.2: The evolution of μ, the distance between inflection points, for the deterministic case with initial condition given by (8.33). This is plotted at various resolutions, illustrating that initially, in the absence of a peakon, μ converges to a non-zero number. As the peakon forms, this measure is no longer converged, and as $\Delta x \to 0$, $\mu \to 0$. Due to errors in measuring the location of the inflection points within a cell, μ can be very noisy. To reduce this, we applied a low pass filter to μ. An interesting feature of the evolution of μ is a fairly sudden drop in the values at around $t = 10$.

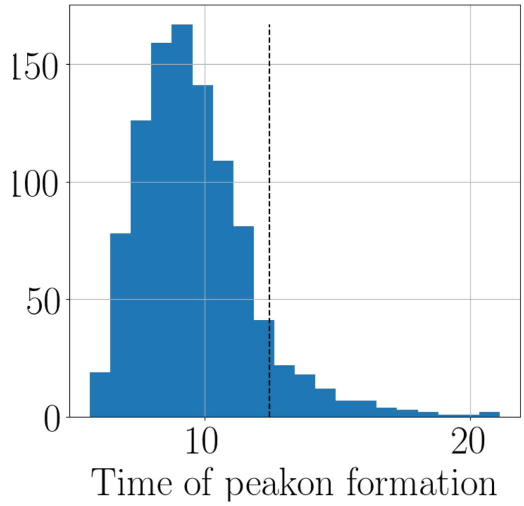


Figure 8.3: A histogram showing the estimated time of peakon formation for 1000 realisations of the noise (8.34) from the initial state (8.33). The dashed black line shows the peakon formation time from the deterministic case. These times are determined to be the first time at which the gradient of the linear best fit curve of $\log(\mu)$ against $\log(\Delta x)$, at a given time, exceeded a threshold, which we chose to be 0.4. The implication of this plot is that the effect of the noise upon peakon formation is to make it occur earlier.

As a peakon forms, values of μ calculated at different resolutions diverge, as illustrated in Figure 8.2, which shows the evolution of a peakon within the discretisation of Section 8.3 at differing resolutions. This figure was produced in a domain of length $L_d = 40$, with a time step of $\Delta t = 0.001$, and with the initial condition set as

$$u = \frac{1}{\exp\left(x - x_c\right) + \exp\left(x_c - x\right)}, \tag{8.33}$$

with $x_c = 203/15$. We have applied a low-pass filter to the μ values so as to reduce error in the measurement of the positions of the inflection points. This data can be used to approximate a time for the formation of the peakon, which we take to be the time at which the gradient of the best fit line of $\log(\mu)$ as a function of $\log(\Delta x)$ at a given time breaches a certain threshold, which we take to be 0.4.

Extending this approach to stochastic simulations, again using the initial condition (8.33), the histograms showing the evolution of μ over many different realisations (as a function of time t and resolution Δx) are shown in Figure 8.3. This experiment was performed for 1000 realisations of the noise, where the stochastic basis functions were $n = 1, 2, \ldots, 6$ of

$$\Xi^n = \begin{cases} 0.08\sin((n+1)\pi x/L_d), & n \text{ is odd}, \\ 0.08\cos(n\pi x/L_d), & n \text{ is even} \end{cases} \tag{8.34}$$

Figure 8.4 clearly shows that the inflection points move closer together as both time evolves, and $\Delta x \to 0$. The histogram showing the peakon formation times from these realisations of the noise is shown in Figure 8.3.

In all these realisations of the noise, a peakon was adjudged to have formed, providing supporting evidence for the hypothesis that peakons form with a probability of unity in the stochastic Camassa-Holm equation from a smooth initial condition with an inflection point on a negative slope, which is clearly not something that can be explicitly proved experimentally. In fact, the evidence from Figures 8.3 and 8.4 is that the stochastic noise prompts the peakon solution to emerge more quickly than in the deterministic case.

8.6 Summary and Outlook

We have extended the discussion of peakons within the stochastic Camassa-Holm equation begun by [115] and [112]. This first result was a demonstration that peakons satisfying the equations (8.18) derived in [115] do indeed satisfy the stochastic Camassa-Holm equation (8.15) when it is cast in hydrodynamic form. Next, we presented a new finite element discretisation for (8.15). This is a natural choice of discretisation for studying peakons, given that the peakon itself is a weak solution to the Camassa-Holm equation. We then showed numerically that this discretisation converges both strongly in time and also converges to (8.30) as $\Delta x \to 0$. This discretisation then provided a tool with which to investigate the steepening lemma of [112] for the stochastic Camassa-Holm system. Starting with smooth initial conditions, we investigated the formation of peakons un-

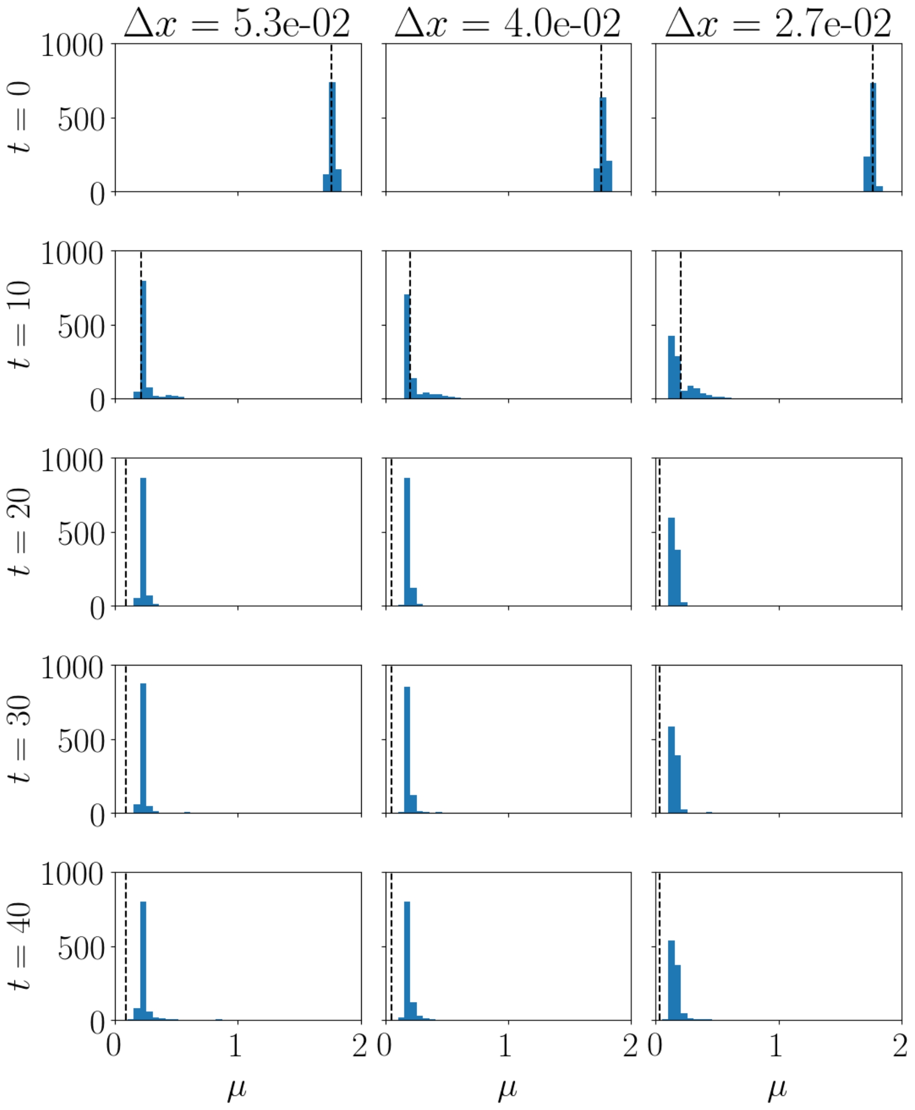


Figure 8.4: Histograms of μ, the distance between inflection points of the profile, over many realisations of the noise. The experiments were all initialised with the condition (8.33), and plotted here are the evolutions of μ with time at different resolutions Δx of the discretisation. In each panel, the dashed black line shows the value of μ in the deterministic case at that resolution. These plots illustrate that μ becomes much smaller as the peakon solution forms from the smooth initial condition. It also shows that as $\Delta x \to 0$, before the peakon forms, μ converges to some non-zero number, but once the peakon has formed it converges to zero. It is also interesting to note that the μ measured in the deterministic case tends to be smaller than in the stochastic cases.

der different realisations of the stochastic noise. No examples without peakon formation were found, and the noise was observed to tend to lead to faster peakon formation.

To summarise, in this chapter we attempted to investigate the behaviour of the Camassa-Holm system when including a stochastic description of unresolved processes according to the formulation of [11]. Our results suggest that the peakon-forming behaviour is still preserved by this approach.

9 Summary and Outlook

Describing the effects of unresolved processes upon the resolved flow is one of the greatest challenges in numerically modelling geophysical fluids. Within many models, the evolution of the resolved flow is treated separately from the parametrisation of the unresolved processes. One major research question is how to couple these components together within the discretisation: both temporally and spatially. At the same time, the use of finite element methods to discretise the equations of geophysical fluids is increasing in prominence. This is due to their mathematical elegance and flexibility over arbitrary meshes, with software developments making these methods more accessible to researchers.

The UK Met Office is developing a compatible finite element discretisation for the resolved flow (the *dynamical core*) of its numerical weather prediction model, which is an important moment in the field. This followed the demonstration by [8] that these methods could provide many of the properties of [9], which are considered important for dynamical cores, while at the same time supporting a quasi-uniform grid over the globe. A new choice of spatial discretisation raises more questions as to the coupling between the resolved and unresolved processes.

Another consideration in describing the effects of the unresolved processes is their uncertainty, which can be captured by the use of stochastic parametrisations. A new approach to this is through the variational principle of [11] in which the transport has a stochastic component, thus offering a new way of coupling together the resolved and unresolved flow.

The work presented in this thesis builds on research in these two areas. Motivated by the decomposition of fields into mean and fluctuating parts, as is common in geophysical parametrisations, Chapter 4 described an implementation of the lowest-order set of compatible spaces in discretising the (dry) compressible Euler equations. The main focus of Chapter 4 was the introduction of a new transport scheme which facilitated the use of these spaces within Gusto, a dynamical core library using a compatible finite element discretisation. This centred on the concept of recovering the fields into a higher-order space, where the transport could take place.

Following this, Chapter 5 presented a method of coupling the compatible finite element dynamical core to physics parametrisations via the inclusion of moisture. Some of the concepts involved in the coupling of these parts of the model were discussed before the model was demonstrated through a series of test cases. Two of these test cases were new, with one being appropriate to use for convergence tests of the full model.

In Chapters 7 and 8 the impact of the stochastic formulation of [11] upon two simple geophysical fluids was investigated. Chapter 7 considered the statistical properties of a finite element discretisation of the stochastic quasi-geostrophic equation. This showed that the choice of discretisation (and the quantities it preserves) can determine the statistics of the stochastic model. The peakon solutions of the stochastic Camassa-Holm equation were the subject of Chapter 8, which presented a new finite element discretisation which converges to the stochastic equations describing the evolution of peakon solutions. This discretisation was used to investigate the formation of peakons within the stochastic equation.

Both of these topics are active areas of future research. As LFRic (the compatible finite element dynamical core of the UK Met Office) is developed, important choices will need to be made. What is the most efficient strategy for the solver? Which set of compatible finite element spaces should be used? Meanwhile there are aims to implement more capabilities in Gusto: flow over topography, a varied choice in temporal discretisations and an arsenal of physical parametrisations to name but a few. Discussed at the end of Chapter 4 were some of the features necessary to the recovered space transport scheme to be used over spherical domains. More generally the coupling of physical parametrisations and dynamical components of numerical weather prediction models has many open questions: how tightly to couple the components (for different physical processes); whether to perform the components upon different spatial grids; how to separately discretise the components in time.

The recent work of [123] upon the stochastic variational framework of [11] has demonstrated that it is promising for use in data assimilation. Future research is needed into how the noise should be distributed spatially through the choice of the Ξ functions. Another related stochastic formulation being explored involves a different choice of the advecting field. In this case, the advecting velocity is written in terms of the Lagrangian-averaged velocity:

$$\mathrm{d}\boldsymbol{x}_t = \mathbb{E}(u)\,\mathrm{d}t + \sum_j \Xi^j \circ \mathrm{d}W, \tag{9.1}$$

where $\mathbb{E}(u)$ satisfies its own deterministic equation featuring the Ξ_j functions.

This thesis presented a collection of works relating to the coupling of resolved and unresolved physical processes in discretisations of geophysical fluids. This is an underexplored field with many open questions, making it a very exciting area of research for the future.

BIBLIOGRAPHY

[1] G. K. Vallis, "Geophysical fluid dynamics: whence, whither and why?," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 472, no. 2192, 2016.

[2] M. Tiedtke, "The general problem of parametrization." ECMWF Meteorological Training Course Lecture Series, 2002.

[3] I. Van der Hoven, "Power spectrum of horizontal wind speed in the frequency range from 0.0007 to 900 cycles per hour," *Journal of Meteorology*, vol. 14, no. 2, pp. 160–164, 1957.

[4] T.-C. Chen and A. Wiin-Nielsen, "On nonlinear cascades of atmospheric energy and enstrophy in a two-dimensional spectral index," *Tellus*, vol. 30, no. 4, pp. 313–322, 1978.

[5] P. R. Gent and J. C. Mcwilliams, "Isopycnal mixing in ocean circulation models," *Journal of Physical Oceanography*, vol. 20, no. 1, pp. 150–155, 1990.

[6] W. G. Large, J. C. McWilliams, and S. C. Doney, "Oceanic vertical mixing: A review and a model with a nonlocal boundary layer parameterization," *Reviews of Geophysics*, vol. 32, no. 4, pp. 363–403, 1994.

[7] S. B. Pope, *Turbulent flows*. IOP Publishing, 2001.

[8] C. J. Cotter and J. Shipton, "Mixed finite elements for numerical weather prediction," *Journal of Computational Physics*, vol. 231, no. 21, pp. 7076–7091, 2012.

[9] A. Staniforth and J. Thuburn, "Horizontal grids for global weather and climate prediction models: a review," *Quarterly Journal of the Royal Meteorological Society*, vol. 138, no. 662, pp. 1–26, 2012.

[10] T. Melvin, T. Benacchio, B. Shipway, N. Wood, J. Thuburn, and C. Cotter, "A mixed finite-element, finite-volume, semi-implicit discretization for atmospheric dynamics: Cartesian geometry," *Quarterly Journal of the Royal Meteorological Society*, 2019.

[11] D. D. Holm, "Variational principles for stochastic fluid dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 471, no. 2176, 2015.

[12] T. M. Bendall and C. J. Cotter, "Statistical properties of an enstrophy conserving finite element discretisation for the stochastic quasi-geostrophic equation," *Geophysical & Astrophysical Fluid Dynamics*, pp. 1–14, 2018.

[13] T. M. Bendall, C. J. Cotter, and J. Shipton, "The recovered space advection scheme for lowest-order compatible finite element methods," *Journal of Computational Physics*, vol. 390, pp. 342–358, 2019.

[14] T. M. Bendall, T. H. Gibson, J. Shipton, C. J. Cotter, and B. Shipway, "A compatible finite element discretisation for the moist compressible Euler equations," *arXiv preprint arXiv 1910.01857*, 2019.

[15] T. M. Bendall, C. J. Cotter, and D. D. Holm, "Perspectives on the formation of peakons in the stochastic Camassa-Holm equation," *arXiv preprint 1910.03018*, 2019.

[16] O. Zienkiewicz, "The birth of the finite element method and of computational mechanics," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 1, pp. 3–10, 2004.

[17] R. W. Clough, "Early history of the finite element method from the view point of a pioneer," *International Journal for Numerical Methods in Engineering*, vol. 60, no. 1, pp. 283–287, 2004.

[18] S. Brenner and R. Scott, *The mathematical theory of finite element methods*, vol. 15. Springer Science & Business Media, 2007.

[19] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, *The finite element method: its basis and fundamentals*. Butterworth-Heinemann, 7th edition. ed., 2013.

[20] P. G. Ciarlet, *The finite element method for elliptic problems*. SIAM, 2002.

[21] D. N. Arnold, D. Boffi, and F. Bonizzoni, "Finite element differential forms on curvilinear cubic meshes and their approximation properties," *Numerische Mathematik*, vol. 129, no. 1, pp. 1–20, 2015.

[22] A. T. McRae, G.-T. Bercea, L. Mitchell, D. A. Ham, and C. J. Cotter, "Automated generation and symbolic manipulation of tensor product finite elements," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S25–S47, 2016.

[23] P.-A. Raviart and J.-M. Thomas, "A mixed finite element method for 2-nd order elliptic problems," in *Mathematical aspects of finite element methods*, pp. 292–315, Springer, 1977.

[24] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, vol. 15. Springer Science & Business Media, 2012.

[25] J. D. Lambert, *Numerical methods for ordinary differential systems: the initial value problem*. John Wiley & Sons, Inc., 1991.

[26] E. Hairer, C. Lubich, and G. Wanner, *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, vol. 31. Springer Science & Business Media, 2006.

[27] B. Cockburn and C.-W. Shu, "Runge–Kutta discontinuous Galerkin methods for convection-dominated problems," *Journal of Scientific Computing*, vol. 16, no. 3, pp. 173–261, 2001.

[28] C.-W. Shu and S. Osher, "Efficient implementation of essentially non-oscillatory shock-capturing schemes," *Journal of Computational Physics*, vol. 77, no. 2, pp. 439–471, 1988.

[29] J. P. Fortney, *A visual introduction to differential forms and calculus on manifolds*, vol. 337. Springer, 2018.

[30] D. N. Arnold, R. S. Falk, and R. Winther, "Finite element exterior calculus, homological techniques, and applications," *Acta numerica*, vol. 15, pp. 1–155, 2006.

[31] H. Brezis, *Functional analysis, Sobolev spaces and partial differential equations*. Springer Science & Business Media, 2010.

[32] D. N. Arnold and A. Logg, "Periodic table of the finite elements," *SIAM News*, vol. 47, no. 9, p. 212, 2014.

[33] D. N. Arnold, "Differential complexes and numerical stability," *arXiv preprint math/0212391*, 2002.

[34] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. McRae, G.-T. Bercea, G. R. Markall, and P. H. Kelly, "Firedrake: automating the finite element method by composing abstractions," *ACM Transactions on Mathematical Software (TOMS)*, vol. 43, no. 3, p. 24, 2017.

[35] A. Logg, K.-A. Mardal, and G. Wells, *Automated solution of differential equations by the finite element method: The FEniCS book*, vol. 84. Springer Science & Business Media, 2012.

[36] M. S. Alnæs, A. Logg, K. B. Ølgaard, M. E. Rognes, and G. N. Wells, "Unified form language: A domain-specific language for weak formulations of partial differential equations," *ACM Transactions on Mathematical Software (TOMS)*, vol. 40, no. 2, p. 9, 2014.

[37] M. Homolya, L. Mitchell, F. Luporini, and D. A. Ham, "TSFC: a structure-preserving form compiler," *SIAM Journal on Scientific Computing*, vol. 40, no. 3, pp. C401–C428, 2018.

[38] M. Homolya, R. C. Kirby, and D. A. Ham, "Exposing and exploiting structure: optimal code generation for high-order finite element methods," *arXiv preprint arXiv:1711.02473*, 2017.

[39] R. C. Kirby and L. Mitchell, "Code generation for generally mapped finite elements," *arXiv preprint arXiv:1808.05513*, 2018.

[40] T. H. Gibson, L. Mitchell, D. A. Ham, and C. J. Cotter, "Slate: extending Firedrake's domain-specific abstraction to hybridized solvers for geoscience and beyond," *arXiv preprint arXiv:1802.00303*, 2019.

[41] G.-T. Bercea, A. T. McRae, D. A. Ham, L. Mitchell, F. Rathgeber, L. Nardi, F. Luporini, and P. H. Kelly, "A structure-exploiting numbering algorithm for finite elements on extruded meshes, and its performance evaluation in Firedrake," *arXiv preprint arXiv:1604.05937*, 2016.

[42] M. Homolya and D. A. Ham, "A parallel edge orientation algorithm for quadrilateral meshes," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S48–S61, 2016.

[43] F. Rathgeber, *Productive and efficient computational science through domain-specific abstractions*. PhD thesis, Imperial College London, 2014.

[44] A. Staniforth, T. Melvin, and N. Wood, "Gungho! a new dynamical core for the unified model," in *Proceeding of the ECMWF workshop on recent developments in numerical methods for atmosphere and ocean modelling*, 2013.

[45] A. Staniforth and N. Wood, "Aspects of the dynamical core of a nonhydrostatic, deep-atmosphere, unified weather and climate-prediction model," *Journal of Computational Physics*, vol. 227, no. 7, pp. 3445–3464, 2008.

[46] J. Thuburn, "Some basic dynamics relevant to the design of atmospheric model dynamical cores," in *Numerical Techniques for Global Atmospheric Models*, pp. 3–27, Springer, 2011.

[47] F. J. Winninghoff, *On the adjustment toward a geostrophic balance in a simple primitive-equation model with application to the problem of initialization and objective analysis*. PhD thesis, Department of Meteorology: University of California, Los Angeles, 1968.

[48] F. Mesinger, A. Arakawa, and H. Sundqvist, *Numerical methods used in atmospheric models*, vol. 1. World Meteorological Organization, International Council of Scientific Unions, 1976.

[49] A. Arakawa and V. R. Lamb, "Computational design of the basic dynamical processes of the UCLA general circulation model," *Methods in Computational Physics*, vol. 17, pp. 173–265, 1977.

[50] A. L. Schoenstadt, "A transfer function analysis of numerical schemes used to simulate geostrophic adjustment," *Monthly Weather Review*, vol. 108, no. 8, pp. 1248–1259, 1980.

[51] R. Williams, “On the formulation of finite-element prediction models,” *Monthly Weather Review*, vol. 109, no. 3, pp. 463–466, 1981.

[52] R. C. Wajsowicz, “Free planetary waves in finite-difference numerical models,” *Journal of Physical Oceanography*, vol. 16, no. 4, pp. 773–789, 1986.

[53] J. Thuburn and A. Staniforth, “Conservation and linear Rossby-mode dispersion on the spherical C grid,” *Monthly Weather Review*, vol. 132, no. 2, pp. 641–653, 2004.

[54] J. Thuburn, “Rossby wave dispersion on the C-grid,” *Atmospheric Science Letters*, vol. 8, no. 2, pp. 37–42, 2007.

[55] D. Y. Le Roux, A. Staniforth, and C. A. Lin, “Finite elements for shallow-water equation ocean models,” *Monthly Weather Review*, vol. 126, no. 7, pp. 1931–1951, 1998.

[56] J. M. Dennis and R. D. Loft, *Refactoring Scientific Applications for Massive Parallelism*, pp. 539–556. Springer Berlin Heidelberg, 2011.

[57] N. Wood, “Personal communication.”

[58] B. N. Lawrence, M. Rezny, R. Budich, P. Bauer, J. Behrens, M. Carter, W. Deconinck, R. Ford, C. Maynard, S. Mullerworth, *et al.*, “Crossing the chasm: how to develop weather and climate models for next generation computers?,” *Geoscientific Model Development*, vol. 11, no. 5, pp. 1799–1821, 2018.

[59] A. Staniforth, T. Melvin, and C. Cotter, “Analysis of a mixed finite-element pair proposed for an atmospheric dynamical core,” *Quarterly Journal of the Royal Meteorological Society*, vol. 139, no. 674, pp. 1239–1254, 2013.

[60] C. J. Cotter and J. Thuburn, “A finite element exterior calculus framework for the rotating shallow-water equations,” *Journal of Computational Physics*, vol. 257, pp. 1506–1526, 2014.

[61] A. T. McRae and C. J. Cotter, “Energy- and enstrophy-conserving schemes for the shallow-water equations, based on mimetic finite elements,” *Quarterly Journal of the Royal Meteorological Society*, vol. 140, no. 684, pp. 2223–2234, 2014.

[62] A. Natale, J. Shipton, and C. J. Cotter, “Compatible finite element spaces for geophysical fluid dynamics,” *Dynamics and Statistics of the Climate System*, vol. 1, no. 1, 2016.

[63] H. Yamazaki, J. Shipton, M. J. Cullen, L. Mitchell, and C. J. Cotter, “Vertical slice modelling of nonlinear Eady waves using a compatible finite element method,” *Journal of Computational Physics*, vol. 343, pp. 130–149, 2017.

[64] J. Shipton, T. Gibson, and C. Cotter, "Higher-order compatible finite element schemes for the nonlinear rotating shallow water equations on the sphere," *Journal of Computational Physics*, vol. 375, pp. 1121–1137, 2018.

[65] T. Melvin, T. Benacchio, J. Thuburn, and C. Cotter, "Choice of function spaces for thermodynamic variables in mixed finite-element methods," *Quarterly Journal of the Royal Meteorological Society*, vol. 144, no. 712, pp. 900–916, 2018.

[66] D. Walters, N. Wood, S. Vosper, and S. Milton, "ENDGame: A new dynamical core for seamless atmospheric prediction," tech. rep., UK Met Office, 2014.

[67] C. J. Cotter and D. Kuzmin, "Embedded discontinuous Galerkin transport schemes with localised limiters," *Journal of Computational Physics*, vol. 311, pp. 363–373, 2016.

[68] A. Iserles, *A first course in the numerical analysis of differential equations*. No. 44, Cambridge University Press, 2009.

[69] D. N. Arnold, "An interior penalty finite element method with discontinuous elements," *SIAM Journal on Numerical Analysis*, vol. 19, no. 4, pp. 742–760, 1982.

[70] F. Brezzi and M. Fortin, *Mixed and hybrid finite element methods*, vol. 15. Springer Science & Business Media, 2012.

[71] E. H. Georgoulis and T. Pryer, "Recovered finite element methods," *Computer Methods in Applied Mechanics and Engineering*, vol. 332, pp. 303–324, 2018.

[72] V. A. Titarev and E. F. Toro, "ADER: Arbitrary high order Godunov approach," *Journal of Scientific Computing*, vol. 17, no. 1-4, pp. 609–618, 2002.

[73] B. Van Leer and S. Nomura, "Discontinuous Galerkin for diffusion," in *17th AIAA Computational Fluid Dynamics Conference*, p. 5108, 2005.

[74] O. A. Karakashian and F. Pascal, "Convergence of adaptive discontinuous Galerkin approximations of second-order elliptic problems," *SIAM Journal on Numerical Analysis*, vol. 45, no. 2, pp. 641–665, 2007.

[75] D. Kuzmin, "A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods," *Journal of Computational and Applied Mathematics*, vol. 233, no. 12, pp. 3077–3085, 2010.

[76] D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber, "A standard test set for numerical approximations to the shallow water equations in spherical geometry," *Journal of Computational Physics*, vol. 102, no. 1, pp. 211–224, 1992.

[77] R. J. LeVeque, "High-resolution conservative algorithms for advection in incompressible flow," *SIAM Journal on Numerical Analysis*, vol. 33, no. 2, pp. 627–665, 1996.

[78] G. H. Bryan and J. M. Fritsch, "A benchmark simulation for moist nonhydrostatic numerical models," *Monthly Weather Review*, vol. 130, no. 12, pp. 2917–2928, 2002.

[79] P. Ullrich, K. Reed, and C. Jablonowski, "Analytical initial conditions and an analysis of baroclinic instability waves in f-and β-plane 3D channel models," *Quarterly Journal of the Royal Meteorological Society*, vol. 141, no. 693, pp. 2972–2988, 2015.

[80] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, and K. K. Droegemeier, "Numerical solutions of a non-linear density current: A benchmark solution and comparisons," *International Journal for Numerical Methods in Fluids*, vol. 17, no. 1, pp. 1–22, 1993.

[81] J. Lander and B. Hoskins, "Believable scales and parameterizations in a spectral transform model," *Monthly Weather Review*, vol. 125, no. 2, pp. 292–303, 1997.

[82] W. C. Skamarock and J. B. Klemp, "Efficiency and accuracy of the Klemp-Wilhelmson time-splitting technique," *Monthly Weather Review*, vol. 122, no. 11, pp. 2623–2630, 1994.

[83] J. Thuburn, "Use of the Gibbs thermodynamic potential to express the equation of state in atmospheric models," *Quarterly Journal of the Royal Meteorological Society*, vol. 143, no. 704, pp. 1185–1196, 2017.

[84] S. A. Rutledge and P. Hobbs, "The mesoscale and microscale structure and organization of clouds and precipitation in midlatitude cyclones. VIII: A model for the seeder-feeder process in warm-frontal rainbands," *Journal of the Atmospheric Sciences*, vol. 40, no. 5, pp. 1185–1206, 1983.

[85] O. Tetens, "Uber einige meteorologische Begriffe," *Z. Geophys.*, no. 6, pp. 297–309, 1930.

[86] J. Monteith and M. Unsworth, *Principles of Environmental Physics*. Academic Press, 2007.

[87] J. B. Klemp and R. B. Wilhelmson, "The simulation of three-dimensional convective storm dynamics," *Journal of the Atmospheric Sciences*, vol. 35, no. 6, pp. 1070–1096, 1978.

[88] S.-T. Soong and Y. Ogura, "A comparison between axisymmetric and slab-symmetric cumulus cloud models," *Journal of the Atmospheric Sciences*, vol. 30, no. 5, pp. 879–893, 1973.

[89] J. Milbrandt and R. McTaggart-Cowan, "Sedimentation-induced errors in bulk microphysics schemes," *Journal of the Atmospheric Sciences*, vol. 67, no. 12, pp. 3931–3948, 2010.

[90] K. A. Emanuel, *Atmospheric convection*. Oxford University Press on Demand, 1994.

[91] I. R. Paluch, "The entrainment mechanism in Colorado cumuli," *Journal of the Atmospheric Sciences*, vol. 36, no. 12, pp. 2467–2478, 1979.

[92] W. W. Grabowski and T. L. Clark, "Cloud–environment interface instability: Rising thermal calculations in two spatial dimensions," *Journal of the Atmospheric Sciences*, vol. 48, no. 4, pp. 527–546, 1991.

[93] F. Bouchut, J. Lambaerts, G. Lapeyre, and V. Zeitlin, "Fronts and nonlinear waves in a simplified shallow-water model of the atmosphere with moisture and convection," *Physics of Fluids*, vol. 21, no. 11, 2009.

[94] M. Zerroukat and T. Allen, "A moist Boussinesq shallow water equations set for testing atmospheric models," *Journal of Computational Physics*, vol. 290, pp. 55–72, 2015.

[95] D. R. Thatcher, "A moist aquaplanet variant of the Held-Suarez test for atmospheric model dynamical cores," *Geoscientific Model Development*, vol. 9, no. 4, p. 1263, 2016.

[96] P. Bauer, A. Thorpe, and G. Brunet, "The quiet revolution of numerical weather prediction," *Nature*, vol. 525, no. 7567, p. 47, 2015.

[97] T. N. Palmer and P. D. Williams, "Introduction. Stochastic physics and climate modelling," *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1875, pp. 2419–2425, 2008.

[98] C. L. Franzke, T. J. O'Kane, J. Berner, P. D. Williams, and V. Lucarini, "Stochastic climate theory and modeling," *Wiley Interdisciplinary Reviews: Climate Change*, vol. 6, no. 1, pp. 63–78, 2015.

[99] C. Penland and B. D. Ewald, "On modelling physical systems with stochastic models: diffusion versus Lévy processes," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 366, no. 1875, pp. 2455–2474, 2008.

[100] E. Mémin, "Fluid flow dynamics under location uncertainty," *Geophysical & Astrophysical Fluid Dynamics*, vol. 108, no. 2, pp. 119–146, 2014.

[101] T. D. Drivas and D. D. Holm, "Circulation and energy theorem preserving stochastic fluids," *arXiv preprint arXiv:1808.05308*, 2018.

[102] N. Van Kampen, "Itô versus Stratonovich," *Journal of Statistical Physics*, vol. 24, no. 1, pp. 175–187, 1981.

[103] K. Jacobs, *Stochastic processes for physicists: understanding noisy systems*. Cambridge University Press, 2010.

[104] P. E. Kloeden and E. Platen, *Numerical Solution of Stochastic Differential Equations*. Springer, 1999.

[105] C. J. Cotter and D. D. Holm, "Continuous and discrete Clebsch variational principles," *Foundations of Computational Mathematics*, vol. 9, no. 2, pp. 221–242, 2009.

[106] D. D. Holm, "Hamiltonian balance equations," *Physica D: Nonlinear Phenomena*, vol. 98, no. 2-4, pp. 379–414, 1996.

[107] D. D. Holm and V. Zeitlin, "Hamilton's principle for quasigeostrophic motion," *Physics of fluids*, vol. 10, no. 4, pp. 800–806, 1998.

[108] D. D. Holm, *Geometric mechanics: Dynamics and symmetry*. Imperial College Press, 2011.

[109] D. D. Holm, J. E. Marsden, and T. S. Ratiu, "The Euler–Poincaré equations and semidirect products with applications to continuum theories," *Advances in Mathematics*, vol. 137, no. 1, pp. 1–81, 1998.

[110] D. D. Holm, T. Schmah, and C. Stoica, *Geometric mechanics and symmetry: from finite to infinite dimensions*. No. 12, Oxford University Press, 2009.

[111] C. J. Cotter, G. A. Gottwald, and D. D. Holm, "Stochastic partial differential fluid equations as a diffusive limit of deterministic Lagrangian multi-time dynamics," *Proc. R. Soc. A*, vol. 473, no. 2205, p. 20170388, 2017.

[112] D. Crisan and D. D. Holm, "Wave breaking for the stochastic Camassa–Holm equation," *Physica D: Nonlinear Phenomena*, vol. 376, pp. 138–143, 2018.

[113] D. Alonso-Orán and A. B. de León, "Stability, well-posedness and blow-up criterion for the incompressible slice model," *Physica D: Nonlinear Phenomena*, 2018.

[114] C. Cotter, D. Crisan, D. D. Holm, W. Pan, and I. Shevchenko, "Modelling uncertainty using circulation-preserving stochastic transport noise in a 2-layer quasi-geostrophic model," *arXiv preprint arXiv:1802.05711*, 2018.

[115] D. D. Holm and T. M. Tyranowski, "Variational principles for stochastic soliton dynamics," *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 472, no. 2187, 2016.

[116] T. M. Bendall, "Stochastic quasi-geostrophic equations: Numerical schemes and statistical properties," Master's thesis, Imperial College London, 2016.

[117] S. Dubinkina and J. Frank, “Statistical mechanics of Arakawa’s discretizations,” *Journal of Computational Physics*, vol. 227, no. 2, pp. 1286–1305, 2007.

[118] G. K. Vallis, *Atmospheric and oceanic fluid dynamics: fundamentals and large-scale circulation*. Cambridge University Press, 2006.

[119] A. Majda and X. Wang, *Nonlinear dynamics and statistical theories for basic geophysical flows*. Cambridge University Press, 2006.

[120] R. Salmon, G. Holloway, and M. C. Hendershott, “The equilibrium statistical mechanics of simple quasi-geostrophic models,” *Journal of Fluid Mechanics*, vol. 75, no. 04, pp. 691–703, 1976.

[121] S. Reich and C. Cotter, *Probabilistic forecasting and Bayesian data assimilation*. Cambridge University Press, 2015.

[122] R. Camassa and D. D. Holm, “An integrable shallow water equation with peaked solitons,” *Physical Review Letters*, vol. 71, no. 11, p. 1661, 1993.

[123] C. Cotter, D. Crisan, D. D. Holm, W. Pan, and I. Shevchenko, “Numerically modeling stochastic Lie transport in fluid dynamics,” *Multiscale Modeling & Simulation*, vol. 17, no. 1, pp. 192–232, 2019.

Appendix: Table of Physical Constants

Specific heat capacity of dry air at const. volume	c_{vd}	717 J kg^{-1} K^{-1}
Specific heat capacity of dry air at const. pressure	c_{pd}	1004.5 J kg^{-1} K^{-1}
Specific heat capacity of water vapour at const. volume	c_{vv}	1424 J kg^{-1} K^{-1}
Specific heat capacity of water vapour at const. pressure	c_{pv}	1885 J kg^{-1} K^{-1}
Specific heat capacity of liquid water at const. pressure	c_{pl}	4186 J kg^{-1} K^{-1}
Specific heat capacity of moist air at const. volume	c_{vml}	$c_{vd} + r_v c_{vv} + (r_c + r_r)c_{pl}$
Specific heat capacity of moist air at const. pressure	c_{pml}	$c_{pd} + r_v c_{pv} + (r_c + r_r)c_{pl}$
Specific gas constant for dry air	R_d	287 J kg^{-1} K^{-1}
Specific gas constant for water vapour	R_v	461 J kg^{-1} K^{-1}
Specific gas constant for moist air	R_m	$R_d + r_v R_v$
Reference latent heat of vaporisation of water at T_R	L_{vR}	2.5×10^6 J kg^{-1}
Latent heat of vaporisation of water	L_v	$L_{vR} - (c_{pl} - c_{pv})(T - T_R)$
Reference temperature	T_R	273.15 K
Reference pressure	p_R	10^5 Pa
Constant in Tetens' formula	C_0^{sat}	610.9 Pa
Constant in Tetens' formula	C_1^{sat}	-17.27
Constant in Tetens' formula	C_2^{sat}	35.86 K
Ratio of R_d to R_v	ϵ	0.623
Ratio of R_d to c_{pd}	κ	2/7