Energy conserving compatible finite element methods for numerical weather prediction

a thesis presented for the degree of

Doctor of Philosophy of Imperial College London

and the

Diploma of Imperial College

by

Golo Albert Wimmer

Department of Mathematics

Imperial College

180 Queen's Gate, London SW7 2BZ

November 2020

I certify that this thesis, and the research to which it refers, are the product of my own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referencing practices of the discipline.

Copyright

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

Thesis advisors: Prof Colin J. Cotter, Dr Werner Bauer

Energy conserving compatible finite element methods for numerical weather prediction

Abstract

The compatible finite element method has recently gained an increased interest in numerical weather prediction, as it allows for higher order discretisations and more general meshes, thus avoiding the parallel computing issues associated with grid poles. Further, it can be seen as a finite element extension of the Arakawa finite difference C grid. A dynamical core based on it is currently in development at the UK Met Office, due to replace the current finite difference latitude longitude grid discretisation.

In this thesis, we focus on compatible finite element discretisations that are energy conserving, which is an important property in particular for climate simulations. This is achieved using a Hamiltonian framework, where energy conservation is expressed via the antisymmetry of a Poisson bracket that underlies the governing equations. In order to achieve an improved field development, we newly incorporate into the Poisson bracket Discontinuous Galerkin upwind and Streamline Upwind Petrov Galerkin stabilisation methods for the thermal, density related, and vorticity field equations. The energy conserving property is validated by coupling the upwind-stabilised spatial discretisations to an energy conserving time discretisation. Further, the discretisations are demonstrated to lead to an improved field development with respect to stability for the 2D and 3D dry compressible Euler, rotating shallow water, and thermal rotating shallow water equations.

Acknowledgments

I would like to thank my supervisors Colin and Werner for their strong support throughout the PhD. Their continuous stream of ideas and our frequent, fruitful discussions have been an invaluable contribution to the work presented in this thesis.

I would also like to thank my family and Lekha for their continued support and love I have received in the past years, which has served as a great foundation for an enjoyable time in my PhD.

Finally, I would like to acknowledge support from the EPSRC Mathematics of Planet Earth Centre for Doctoral Training at Imperial College London and the University of Reading, which enabled me to pursue the PhD.

Contents

0 Introduction **1**
- 0.1 Motivation . . . 1
- 0.2 Thesis overview . . . 3
- 0.3 Publications and novel contributions . . . 5

1 Preliminaries **6**
- 1.1 Compatible finite element methods . . . 6
  - 1.1.1 Motivation . . . 7
  - 1.1.2 Formulation . . . 10
  - 1.1.3 Properties . . . 19
  - 1.1.4 Thermal field spaces . . . 23
- 1.2 Hamiltonian framework for energy conservation . . . 24
  - 1.2.1 Rotating shallow water equations . . . 28
  - 1.2.2 Thermal rotating shallow water equations . . . 29
  - 1.2.3 Euler equations . . . 30
  - 1.2.4 Discretisation . . . 33
- 1.3 Upwind stabilisation methods . . . 36
  - 1.3.1 Classical DG upwind stabilisation . . . 37
  - 1.3.2 Lie derivative type upwinding . . . 38
  - 1.3.3 SUPG stabilisation . . . 39
  - 1.3.4 Other types of stabilisation . . . 41

2 Density stabilisation **46**
- 2.1 DG upwind stabilisation in the discretised bracket . . . 47
  - 2.1.1 Flux dependent formulation . . . 49
  - 2.1.2 Velocity upwinding . . . 50
- 2.2 Numerical results . . . 53
  - 2.2.1 Nonlinear system . . . 54
  - 2.2.2 Comparison to other discretisations . . . 58
  - 2.2.3 Test cases . . . 59
- 2.3 Summary . . . 70

3 Thermal field stabilisation **71**
- 3.1 SUPG stabilisation in the discretised bracket . . . 72
- 3.2 Time discretisation . . . 78
  - 3.2.1 Fully energy conserving formulation and nonlinear system . . . 79
  - 3.2.2 Fully upwinded, approximately energy conserving formulation . . . 80
- 3.3 Numerical results . . . 83

3.3.1 Comparison to non-energy conserving space discretisations 83
3.3.2 Test cases 85
3.4 Summary 93

4 Vorticity stabilisation **94**
4.1 SUPG stabilisation in the discretised bracket 95
4.1.1 Thermal two-dimensional case 99
4.1.2 Thermal three-dimensional case 103
4.2 Time discretisation 108
4.2.1 Fully discretised schemes 113
4.3 Numerical results 115
4.3.1 Test cases 116
4.4 Summary 121

5 Conclusion **127**

Appendix A Finite element cell mappings **130**
A.1 Rehabilitation 134

Appendix B Hamiltonian formulation based on momentum **136**
B.1 Preliminaries 136
B.2 Lie-Poisson bracket formulation 145

Appendix C Illustration of upwind stabilisation **152**
C.1 Motivation 152
C.2 Classical DG upwind method 155
C.3 SUPG method 157
C.3.1 Choice of stabilisation parameter 159

Appendix D 3D vorticity formulation with boundaries **163**

References **177**

0 Introduction

0.1 Motivation

The compatible finite element approach has recently been proposed as a discretisation method for numerical weather prediction (NWP) [CS12]. It relies on the use of so-called de Rham complexes of finite element spaces, where one space is mapped to another via differential operators. This leads to desirable stability and convergence properties and, in the context of weather prediction, further allows the use of pseudo-uniform grids on the sphere that avoid the parallel computing issues associated with the latitude-longitude grid. These issues arise due to a relatively fine mesh resolution towards the grid poles, which in turn requires more communication between mesh cells during each time step [FGH+13]. Moreover, the compatible finite element method is quite general, allowing for an unstructured, variable mesh refinement and higher-order discretisations. In view of NWP, it can be seen as a finite element analogue of the Arakawa finite difference C grid method, which is used in the current dynamical core – that is the atmosphere model's fluid dynamics component – of the UK Meteorological Office. For this reason, and in order to achieve a higher parallel computing efficiency, the UK Met Office's next generation dynamical core will be based on a cube sphere grid compatible mixed finite element discretisation.

An important aspect of discretisations in NWP, particularly for climate simulations, is conservation of quantities such as mass and energy. While the former can be conserved using a suitable discretisation of the continuity equation, the latter requires a careful

discretisation of all prognostic equations, ensuring that the energy losses and gains are balanced between the discretised terms. A failure to do so may lead to unbalanced transfers between kinetic, potential, and internal energy, which in turn may led to energy biases in climate models [LR11]. Two ways of guiding this discretisation process include a variational approach (see e.g. [BGB19, NC17b]) and a Hamiltonian framework. Unlike the former, the latter allows for a more direct control of vorticity up to baroclinic terms. A description of the compressible Euler equations within the Hamiltonian framework was first presented in [MG80] in the setting of magnetohydrodynamics, and relies on a formulation where the equations are inferred by a Poisson bracket. This allows for the use of non-canonical (and hence in this case physical) variables [She90], and many fluid dynamical equations have since been formulated in Poisson bracket form [Mor82, Mor98]. Conservation of energy follows easily from this setup via the bracket's antisymmetry, and will be maintained in any space discretisation that preserves the latter. In the context of NWP, the Poisson bracket framework has already been considered for example in [Sal04] and [GH08] for finite difference discretisations. Further, for compatible finite element methods, it has first been considered in [MC14] for the rotating shallow water equations and by extension for the sphere and hemisphere with boundaries in [LP18] and [BC18], respectively. For the compressible Euler equations, which form the basic equation set of dynamical cores in NWP, Hamiltonian based formulations also exist e.g. for the compatible finite element method [LP20], and for general classes of mimetic discretisations [TGS+20].

In particular, this framework facilitates the construction of energy conserving higher order advection methods. Advection terms are an integral part of the governing equations used in NWP, and for finite element discretisations, the methods to be used depend on the underlying finite element spaces. For mixed finite element methods, this requires the application of different methods for different fields. These methods should be consistent (with respect to the continuous equations) and stable, while avoiding an excessive use of diffusion to achieve the latter. Two classical examples of such methods are the standard Discontinuous Galerkin upwind stabilisation method [CKS00, Kuz10] and the Streamline Upwind Petrov-Galerkin (SUPG) stabilisation method [BH82] for discontinuous and continuous Galerkin finite element spaces, respectively. To implement such advection methods while maintaining energy conservation, first, bracket terms related to advection are modified to include the desired advection method, then the remaining bracket terms are adjusted to retain antisymmetry and therefore energy conservation. Within the Poisson bracket framework, DG upwind stabilisation for the buoyancy field in the thermal

shallow water equations has already been introduced in [EDK19]. Further, an energy conserving upwind stabilisation method for velocity advection was presented in a variational framework via Lie derivatives for the incompressible Euler equations in [NC17b], which was also considered for the shallow water case in [NSC16]. Finally, an energy conserving SUPG method was considered for potential vorticity advection in the shallow water equations in [BC18], exploiting the fact that the Poisson bracket term corresponding to velocity advection is antisymmetric in itself. These energy conserving upwind stabilised advection methods can in principle, with the exception of the potential vorticity method, be implemented readily for the three-dimensional dry compressible Euler equations within a compatible finite element discretisation.

However, such methods are still missing for density advection, vorticity advection in the thermal and 3D case, and for potential temperature advection if a continuous (or partially continuous) finite element space is used for the potential temperature field. In particular, the latter is the case for the finite element equivalent of a Charney-Phillips finite difference grid, where the space's node locations are set to coincide with the velocity space nodes corresponding to the vertical wind [NSC16]. This choice of temperature space was recommended in a dispersion property study in [MBTC18] for mixed finite element methods in NWP when compared to a fully discontinuous or fully continuous finite element space, and will be used in the UK Met Office's next dynamical core. In addition to ensuring transport stabilisation and energy conservation, these methods should also be implementable in practice, without incurring a significant increase in computational cost.

0.2 Thesis overview

In this thesis, we newly introduce formulations for the aforementioned missing three advection methods to arrive at a fully upwind stabilised, energy conserving compatible finite element discretisation for the 3D compressible Euler equations. These will be introduced within one underlying Poisson bracket, which, depending on the choice of Hamiltonian, leads to different sets of governing equations. In view of a comprehensible presentation of the advection methods, this allows us to move along a model hierarchy, given by the rotating shallow water, thermal shallow water, 2D dry compressible Euler, and finally 3D dry compressible Euler equations.

In Chapter 1, we review the main mathematical concepts used thereafter. First, we describe the underlying space discretisation, i.e. the compatible finite element method

and the choice of thermal field finite element space. We then move on to the Hamiltonian framework and introduce the underlying Poisson bracket used in this thesis, and demonstrate how to recover the governing equations appearing in our model hierarchy. The section further includes an outline of the energy conserving time discretisation to be used in the main chapters' numerical results sections. Finally, we review common stabilisation methods used for transport equations.

In Chapter 2, we incorporate classical DG upwinding for the density related field. To do so, we introduce an additional operator $\mathbb{U}$ to recover the velocity field from the momentum flux arising in the Hamiltonian framework. The formulation and results are presented for the rotating shallow water equations, where the density related field is given by the total water depth. The fully discretised scheme is tested for energy conservation and further assessed for its qualitative field development in comparison to a version without upwind stabilisation for the depth field, and a non-energy conserving version including the same type of upwinding.

In Chapter 3, we incorporate the SUPG method for the thermal field transport equation, for which we introduce two additional operators θ^a and $\mathbb{S}_{\mathbf{u}}$, which correspond to thermal field advection and an SUPG type contribution. We further present results for the thermal rotating shallow water as well as 2D compressible Euler equations. For the former, we use a fully continuous thermal field space, while for the latter a Charney-Phillips type potential temperature space is considered. Additionally, in view of the increased computational cost required to achieve energy conservation to machine precision, we also consider a simplified scheme, which relies on an approximation to the time discretisation and a small adjustment in the discretised bracket.

In Chapter 4, we incorporate the SUPG method for the vorticity field, following a similar approach as considered in [BC18]. Next to SUPG for vorticity in the context of the shallow water equations, [BC18] mainly considers the treatment of vorticity in the presence of boundaries, and the framework presented therein is extended to the three-dimensional thermal case in Appendix D. Results are presented for the thermal shallow water as well as 2D and 3D compressible Euler equations. For the latter, both CG and Charney-Phillips type temperature spaces are considered, and a comparison between schemes corresponding to streamline upwind stabilised vorticity and upwind stabilised velocity transport is made.

In Chapter 5, we summarise our results and discuss possible directions for future work.

Finally, we note that next to the aforementioned Appendix D, the thesis contains three additional appendices. Appendix A complements the description of the finite element space discretisation given in Chapter 1, and contains an outline on cell mappings required to construct the spaces considered in the compatible framework. Appendix B offers an alternative formulation of the Hamiltonian framework, using momentum instead of velocity. It describes the equivalence between the so-called Lie-Poisson bracket and the Euler-Poincaré equation, and further includes observations on challenges with respect to the discretisation of the momentum based framework in space, as well as similarities to the framework used in the thesis' main chapters. Finally, Appendix C sheds light on the origin of spurious oscillations that may arise from discretised transport terms, and illustrates their mitigation by means of the classical DG upwind and SUPG stabilisation methods. Further, it contains a discussion on the choice of stabilisation parameter used in the latter method.

0.3 Publications and novel contributions

The thesis' main novel contributions are given by Chapters 2, 3, and 4. The content of Chapter 2 can originally be found in "*Energy conserving upwinded compatible finite element schemes for the rotating shallow water equations*" [WCB20b], while the content of Chapter 3 originates from "*Energy conserving SUPG methods for compatible finite element schemes in numerical weather prediction*" [WCB20a]. Finally, parts of the preliminaries reviewed in Chapter 1 are also based on the aforementioned papers. These parts in turn are founded on established theory in the literature. In Chapters 2, 3, and 4, and their underlying papers, my contribution was in designing, implementing and benchmarking the energy conserving transport methods including upwind stabilisation, with input and guidance from my supervisors/co-authors.

1

Preliminaries

In this chapter, we review the main concepts used in the thesis. The first section introduces the compatible finite element method, and includes properties relevant to its use in numerical weather prediction. Further, we describe the continuous Galerkin and Charney-Phillips thermal field finite element spaces used in the chapters to follow. In the second section, we introduce the Hamiltonian framework used to ensure energy conservation, and describe how to reformulate the governing equations as an ordinary differential equation for functionals. This equation, commonly referred to as a Poisson system, is based on a special bilinear form called a Poisson bracket, which is described in detail and forms the main mathematical tool thereafter. Finally, the last section contains a review of stabilised transport methods. This includes the classical DG upwind as well as SUPG stabilisation methods, which are incorporated into the Poisson bracket framework in Chapters 2, 3, and 4. A mathematical illustration of these two methods' stabilising effect is given in Appendix C.

1.1 Compatible finite element methods

In this section, we describe the compatible finite element method, which is the underlying space discretisation considered in the chapters to follow. First, we motivate and define it by means of the linearised rotating shallow water equations, followed by a review of some of its properties relevant to numerical weather prediction. To motivate the Hamiltonian framework, the review includes a discussion on energy conservation for the full rotating shallow water equations. Finally, we introduce the thermal field spaces used thereafter.

1.1.1 Motivation

In the design of dynamical cores for numerical weather and climate prediction models, a number of properties are considered desirable to achieve a good degree of stability and accuracy of the numerical solution, including [ST12]

1. Mass conservation
2. No geopotential gradient or pressure gradient induced source of vorticity
3. Accurate representation of balanced flow and adjustment
4. Absence of computational modes
5. Energy conservation

Mass and energy conservation are important in particular for climate simulations, and a spurious source of either can lead to spurious motions, systematic biases and other secondary effects [LR11, Thu08]. The gradients of geopotential and pressure vanish in the derivation of the vorticity equation due to the curl-grad identity $\nabla \times \nabla\Phi = 0$ (for scalar field Φ), and the latter should still hold true after discretisation to avoid an unphysical vorticity source term. The Earth's large scale atmospheric flows are dominated by hydrostatic and geostrophic (near-)balance, and an accurate numerical representation of the latter is based on the div-curl identity $\nabla \cdot (\nabla \times \mathbf{v}) = 0$ (for vector field $\mathbf{v}$). If these discrete vector calculus identities do not hold, and the resulting numerical sources are not controlled, spurious fast scale dynamics may pollute the numerical solution over short timescales [ST12]. Finally, computational modes, i.e. wave modes with no continuous counterpart, should be avoided or controlled as their existence can lead to unphysical behaviour in response to forcing by e.g. boundary conditions or nonlinearity. The existence of such modes is related to the ratio of velocity and pressure degrees of freedom, which in the case of finite difference and finite element discretisations in turn depends on the underlying grid as well as staggering and the choice of finite element spaces, respectively [CS12, ST12].

One way to maintain the absence of a geopotential and pressure gradient type term in the vorticity equation is given by the Arakawa finite difference C grid discretisation [AL77]. If the latter is combined with a latitude-longitude grid – which is quadrilateral, orthogonal, and uniform up to the grid poles – the remaining aforementioned properties can also be achieved [ST12]. This type of grid has been among the most common grids in

use for numerical weather prediction [ST12], and is used with an extension to the aforementioned C grid scheme in the current dynamical core of the UK Met Office, ENDGame [MDW+10]. An important artefact of the latitude-longitude grid is given by its two grid pole singularities (see Figure 1.1), where the grid's longitude lines meet. This leads to a much finer resolution near the poles when compared to other regions of the sphere, thus severely restricting the time step size for explicit methods via the CFL condition. To alleviate this problem, semi-implicit semi-Lagrangian schemes have become common in combination with latitude-longitude grids [MDW+10, ST12].

Unfortunately, the grid-pole induced resolution clustering leads to another difficulty related to computational efficiency. Due to heat restrictions on the performance of processing units, modern supercomputers increasingly rely on parallelisation, thus favouring a development towards algorithms with a higher parallel efficiency [FGH+13]. In NWP, parallelisation can be achieved by dividing the domain – i.e. the atmosphere – into sub-domains, each of which is then assigned to separate processors. In order to balance the processors' workloads, an approximately equal number of data points is assigned to each one. In particular, sub-domains at resolution clusters – such as the grid poles in a latitude-longitude grid – are of a smaller size (with respect to the atmosphere) when compared to ones away from such clusters. Next to load balancing, communication between processors also needs to be considered to ensure a high parallel efficiency. Since the solution in each sub-domain also depends on that of the neighbouring ones, data at segments near the sub-domains' boundaries needs to be communicated between the nodes periodically. However, as the sub-domains are smaller near the grid poles, a larger portion of data points falls within such segments there, leading to a higher overall need for communication at the grid poles. This in turn leads to a computational bottleneck that worsens as the resolution is increased [ST12]. In view of ever higher resolutions in future numerical weather prediction models, this provides a strong incentive for moving towards other types of grids, and since the underlying discretisations strongly depend on the choice of grid, this also applies to the choice of discretisation.

One such approach is given in [TRSK09], where the Arakawa C finite difference grid discretisation is extended to more general classes of locally orthogonal meshes, while maintaining a discrete geostrophic balance. It was further extended in [RTKS10] to also retain a discrete (horizontal) curl-grad identity $\nabla \times (\nabla \Phi) = 0$. The resulting scheme, informally known as TRiSK, is used e.g. together with a Voronoi type polygonal grid in the earth-system simulation model MPAS [SKD+12]. Another approach is given by the Ger-

man weather forecasting service's dynamical core ICON, which is based on an icosahedral triangular grid (see Figure 1.1) combined with a C grid discretisation [ZRRB15]. A similar version, called ICON-IAP, also exists for icosahedral hexagonal grids [Gas13]. While the approach in [TRSK09] ensures a discrete geostrophic balance, the one described in [Gas13] is set up to merely keep deviations to it small. In return, it is additionally aimed at avoiding the occurrence of the so-called Hollingsworth instability [HKRB83], which is related to whether the identity $(\mathbf{v}\cdot\nabla)\mathbf{v} = (\nabla\times\mathbf{v})\times\mathbf{v}+\frac{1}{2}|\mathbf{v}|^2$ is satisfied after discretisation.

The choice of grid in combination with the above C grid discretisations is also related to Property 4 above, and has been examined e.g. in [Gas11, TCD14, WTC12]. For instance, cube sphere grids – whose quadrilateral mesh ensures a correct ratio of velocity and pressure degrees of freedom – were found to perform worse than icosahedral grids, possibly due to the particular choice of discretisation of the Coriolis term. In view of computational modes, hexagonal grids give rise to spurious Rossby wave modes. However, their low frequency is similar to high wave number physical modes, and a damaging effect has not been reported in practice [CS12, TCD14]. Irrespectively of such modes, the choice of grid in MPAS and ICON is related to practical mesh refinement based arguments, and the more severe spurious inertia-gravity wave mode occurring in the ICON triangular grid is treated by means of numerical filtering.

Other developments have focused on the finite element method, which allows for a more direct flexibility with respect to the underlying choice of grid. This includes the discontinuous Galerkin and spectral element methods (e.g. [BKN15, TF10]), where in the latter high order finite elements are used to minimise dissipation arising from the stabilisation of discretised terms. Next to avoiding grid poles via allowing for more general meshes, these two methods further focus on computational efficiency, as they lead to block-diagonal matrices and diagonal mass matrices, respectively. Another method, currently pursued by the UK Met Office [FGH+13], is given by the mixed finite element method, where different finite element spaces are used for different fields. In particular, their next dynamical core, GungHo, is based on this method together with a cube sphere grid (see Figure 1.1), ensuring the correct ratio of velocity and pressure degrees of freedom.

Similar to the finite difference case, Properties 2 and 3 above can be realised in a finite element setting by satisfying discretised curl-grad and div-curl identities. Finite element discretisations that satisfy discrete vector calculus identities are known as "mimetic" or "compatible", and have a long history in a range of modelling applications [BBF+13] with a rich

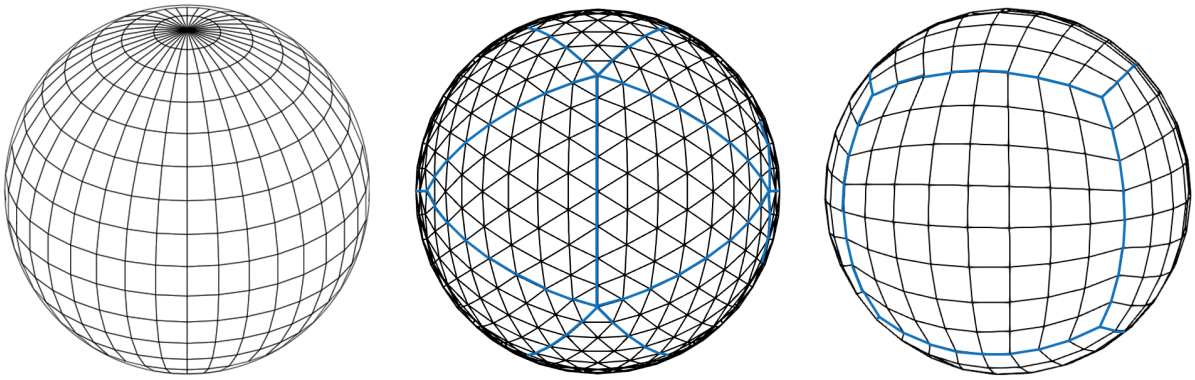


Figure 1.1: Latitude-longitude, icosahedral, and cube sphere grids. Base icosahedron and cube are marked in blue.

underlying theoretical body based on finite element exterior calculus [AFW06, AFW10]. In view of numerical weather forecasting, they have been considered more recently e.g. for the spectral element method [TF10] and mixed element methods [CS12, CM14, NSC16]. The latter can be seen as an extension to the finite difference C grid, and the aforementioned mixed method used in the UK Met Office's next dynamical core is indeed compatible.

The remainder of this thesis is based on the compatible mixed finite element method, and in the section below we define it for sets of finite element spaces used thereafter. In view of the above discussion, we include a remark on the choice of spaces relative to Property 4. Further, Section 1.1.3 includes a brief discussion of Properties 2 and 3 in view of the compatible mixed finite element method and its mimetic features, which will be relied upon in the discretisations thereafter.

1.1.2 Formulation

In this section, we briefly review the finite element spaces used thereafter (following e.g. [BBF+13]), and formulate the compatible mixed finite element method.

As finite element methods are concerned with the discretisation of weak forms of differential equations, we first recall the following standard function spaces to be used below.

Given a closed and bounded domain $\Omega \subset \mathbb{R}^d$, we define

$$L^2(\Omega) \coloneqq \{\psi \colon \Omega \to \mathbb{R} \mid \int_\Omega |\psi|^2 \, dx < \infty\}, \tag{1.1}$$

$$H(\mathrm{div}, \Omega) \coloneqq \{\mathbf{v} \in \left(L^2(\Omega)\right)^d \mid \nabla \cdot \mathbf{v} \in L^2(\Omega)\}, \tag{1.2}$$

$$H(\mathrm{curl}, \Omega) \coloneqq \{\mathbf{v} \in \left(L^2(\Omega)\right)^d \mid \nabla \times \mathbf{v} \in \left(L^2(\Omega)\right)^d\}, \tag{1.3}$$

$$H^1(\Omega) \coloneqq \{\psi \in L^2(\Omega) \mid \nabla\psi \in \left(L^2(\Omega)\right)^d\}, \tag{1.4}$$

the standard L^2 space, Sobolev type spaces in which the divergence and curl operators are well-defined, and the standard Sobolev space H^1. $\nabla\cdot$, $\nabla\times$, and ∇ denote the divergence, curl, and gradient operators, respectively. Functions in $H(\mathrm{div}, \Omega)$ satisfy the divergence theorem, i.e. for $\mathbf{v} \in H(\mathrm{div}, \Omega)$ and any open bounded subset $K \subseteq \Omega$,

$$\int_K \nabla \cdot \mathbf{v} \, dx = \int_{\partial K} \mathbf{v} \cdot \mathbf{n} \, dS, \tag{1.5}$$

for boundary ∂K of K and outward pointing unit vector $\mathbf{n}$ [*]. Similarly, functions in $H(\mathrm{curl}, \Omega)$ satisfy Stokes' theorem, i.e.

$$\int_\Sigma \nabla \times \mathbf{v} \, dx = \int_{\partial\Sigma} \mathbf{v} \cdot \mathbf{n} \, dS, \tag{1.6}$$

for a surface Σ with boundary $\partial\Sigma$.

To motivate the compatible finite element method, we consider the linearised shallow water equations, which form the most basic version of governing equations considered in the chapters to follow. They are given by a momentum and a continuity equation of the form

$$\mathbf{u}_t + f\mathbf{u}^\perp + g\nabla D = 0, \tag{1.7}$$

$$D_t + H\nabla \cdot \mathbf{u} = 0, \tag{1.8}$$

in a (topologically) two-dimensional domain Ω, for flow velocity $\mathbf{u}(\mathbf{x}, t)$ and depth $D(\mathbf{x}, t)$, where $\mathbf{x} \in \Omega$ and $t \geq 0$ denote position in space and time, respectively. The subscript in t denotes differentiation in time, and the superscript $^\perp$ denotes the perpendicular operator, given such that $\mathbf{u}^\perp = \mathbf{k} \times \mathbf{u}$, for the domain's outward unit vector $\mathbf{k}$. In the chapters to follow, for the shallow water equations, Ω will be given by spherical and periodic rectangular domains. Note that for the spherical case, the divergence and

[*] Strictly speaking, $\mathbf{v}$ in the boundary integral is given by its projection $\mathbf{v}|_{\partial\Omega}$ into the trace space $L^2(\partial\Omega)$, and for ease of notation, we drop the $|_\Omega$.

gradient operators are considered tangentially to the domain. The parameters f, g, and H denote the Coriolis parameter, gravitational acceleration, and (constant) mean depth, respectively. We consider a weak form

$$\langle \mathbf{v}, \mathbf{u}_t + f\mathbf{u}^\perp \rangle - \langle \nabla \cdot \mathbf{v}, gD \rangle = 0, \qquad \forall \mathbf{v} \in H(\text{div}, \Omega), \tag{1.9}$$

$$\langle \psi, D_t + H\nabla \cdot \mathbf{u} \rangle = 0 \qquad \forall \psi \in L^2(\Omega), \tag{1.10}$$

for L^2 inner product $\langle ., . \rangle$, test functions $\mathbf{v}$, ψ and solution fields $\mathbf{u} \in H(\text{div}, \Omega)$, $D \in L^2(\Omega)$. Note that the choice of function spaces guarantees that the weak form is well-defined, assuming further that $D_t \in L^2(\Omega)$, $\mathbf{u}_t \in (L^2(\Omega))^2$.

Finite elements

In the finite element method, the weak form is then discretised by considering finite dimensional sub-spaces of the infinite dimensional solution spaces. In what follows, we briefly outline the construction of finite element spaces using finite elements.

Definition 1. *A finite element is a triple $(K, \mathcal{P}, \mathcal{N})$ such that*

i) $K \subset \mathbb{R}^n$ is a simply connected bounded open set with piece-wise smooth boundary

ii) $\mathcal{P}$ is a finite-dimensional space of functions defined on K

iii) $\mathcal{N}$ is a basis of $\mathcal{P}'$

Note that $\mathcal{P}'$ denotes the dual space of $\mathcal{P}$. K is the element domain, $\mathcal{P}$ the space of shape functions, and $\mathcal{N}$ the set of nodal variables (also referred to as degrees of freedom). Given a reference finite element $(\hat{K}, \hat{\mathcal{P}}, \hat{\mathcal{N}})$ and the domain Ω in which the equations are solved, a corresponding finite element space can then be constructed by mapping the finite element to each sub-domain K of a tessellation $\mathcal{T} = \bigcup_{K \subset \Omega} K$ of Ω. Note that implicitly in this construction, we also need to specify which nodes of $\mathcal{N}$ in each cell K are shared with other cells. The sharing of nodes enforces continuity between cells, allowing the construction of conformal finite element spaces, i.e. spaces that are sub-spaces of the corresponding solution spaces. In the chapters below, the following finite elements will be considered (see also Figures 1.3, 1.4).

Continuous Galerkin (CG). The reference element domain $\hat{K}$ is given by an interval or triangle, the shape functions $\hat{\mathcal{P}}$ by the space $\mathcal{P}_k$ of polynomials up to degree $k \geq 1$,

i.e.

$$\mathcal{P}_k = \text{span}\big(\{\hat{x}^a \hat{y}^b \mid (a,\ b) \in \mathbb{N}^2,\ 0 \leq a + b \leq k\}\big), \tag{1.11}$$

for a triangle with reference cell coordinates $(\hat{x}, \hat{y})$, and analogously for intervals. The nodes $\mathcal{N}$ are given by function evaluations at e.g. uniformly spaced points, or points according to a Gauss-Lobatto quadrature in the case of an interval. The nodes corresponding to points on the interval's ends and triangle's vertices and edges are shared. This enforces continuity between elements, therefore leading to a H^1 conforming element. Further, square and box CG elements can be created as tensor products of CG interval elements, with shape function spaces of the form

$$\mathcal{P}_{r,s} = \mathcal{P}_r \otimes \mathcal{P}_s, \tag{1.12}$$

for squares, and similar for boxes. For simplicity, we only consider the case $r = s$, and write CG_k for square/box elements which are tensor products of interval CG_k spaces. Further, we denote triangular elements by P_k.

Discontinuous Galerkin (DG). The finite element is equal to the CG one, except that nodes on each element's boundaries are not shared with other elements. Therefore, there is no enforcement of continuity, and DG spaces are only L^2-conforming. Note that in this case $k = 0$ (and also $r = s = 0$ for squares) also leads to a non-trivial space, given by the space of piece-wise constant functions (rather than fully constant functions in the continuous case). Similarly to the CG case, we write DG_k for square or box elements, and P_k^{DG} for triangular elements.

H(div) conforming. Given functions $\psi \in C^\infty(\Omega)$ and $\mathbf{v} \in H(\text{div}, \Omega)$, we find from the divergence theorem (1.5) applied in each cell K of a tessellation $\mathcal{T}$ of Ω,

$$\begin{aligned}\int_\Omega \psi \nabla \cdot \mathbf{v}\ dx = \sum_{K \in \mathcal{T}} \int_K \psi \nabla \cdot \mathbf{v}\ dx &= \sum_{K \in \mathcal{T}} \Big(-\int_K \nabla\psi \cdot \mathbf{v}\ dx + \int_{\partial K} \psi \mathbf{v} \cdot \mathbf{n}\ dS \Big) \\ &= -\int_\Omega \nabla\psi \cdot \mathbf{v}\ dx + \sum_{K \in \mathcal{T}} \int_{\partial K} \psi \mathbf{v} \cdot \mathbf{n}\ dS. \end{aligned} \tag{1.13}$$

However, the divergence theorem applied in the whole of Ω also leads to

$$\int_\Omega \psi \nabla \cdot \mathbf{v}\ dx = -\int_\Omega \nabla\psi \cdot \mathbf{v}\ dx, \tag{1.14}$$

and subtracting this from (1.13), we find that the sum of facet contributions ∂K is equal to zero. This holds for any given function $\psi \in C^\infty(\Omega)$, and in particular, by picking a

series of test functions ψ_ϵ whose compact support in Ω converges to a given facet ∂K, the sum in (1.13) reduces in the limit to

$$\int_{\partial K} \mathbf{v}^+ \cdot \mathbf{n}^+ \, dS + \int_{\partial K} \mathbf{v}^- \cdot \mathbf{n}^- \, dS = \int_{\partial K} (\mathbf{v}^+ - \mathbf{v}^-) \cdot \mathbf{n}^+ = 0, \tag{1.15}$$

for values $+$, $-$ on either side of ∂K, with $\mathbf{n}^- = -\mathbf{n}^+$ [†]. With this in mind, H(div)-conforming finite element spaces can be created using polynomials in each element and ensuring continuity of normal components of cell facets via the nodal variables.

For triangular tessellations of the domain Ω, we will consider the Brezzi-Douglas-Marini (BDM_k) finite element [BDM85]. The space of shape functions is given by $(\mathcal{P}_k)^2$, for $k \geq 1$. Note that this space has dimension $(k+1)(k+2)$. To ensure the resulting finite element space is $H(\mathrm{div})$-conforming, $k+1$ nodes are used as integral moments of the field's normal component along each of the three triangle's edges; the remaining $(k+1)(k+2) - 3(k+1) = (k-1)(k+1)$ nodes are evaluated as integral moments over the triangle. Noting that in practice the integral moments are evaluated using a quadrature, the nodes can also be formulated as function evaluations at quadrature points instead of moments.

For square shaped cells, $H(\mathrm{div})$-conforming elements can be created using tensor products of interval CG and DG elements [ABF05]. In 2D, for $k \geq 1$, the element is given by

$$H(\mathrm{div})_k = (\hat{\mathbf{x}} CG_k \otimes DG_{k-1}) \oplus (DG_{k-1} \otimes \hat{\mathbf{y}} CG_k)\,, \tag{1.16}$$

where $\hat{\mathbf{x}}$, $\hat{\mathbf{y}}$ correspond to the unit vectors in the $\hat{x}$ and $\hat{y}$ direction. Note that in the tensor product element, the unit vectors are normal to the square's edges, and paired with CG spaces this ensures that the resulting finite element space has continuous normal components (see Figure 1.2). Further, note that this element is analogous to one known in the literature as the Raviart-Thomas element RT_k [BBF+13], and our naming convention is to reflect the tensor product setup in common with the other H(div) and H(curl) type elements considered here. Finally, box elements are created accordingly with another DG space in the product:

$$\begin{aligned} H(\mathrm{div})_k = (\hat{\mathbf{x}} CG_k \otimes DG_{k-1} \otimes DG_{k-1}) \oplus (DG_{k-1} \otimes \hat{\mathbf{y}} CG_k \otimes DG_{k-1}) \\ \oplus (DG_{k-1} \otimes DG_{k-1} \otimes \hat{\mathbf{z}} CG_k)\,, \end{aligned} \tag{1.17}$$

[†] For tessellations of the sphere, this need not hold exactly; for a discussion of this problem see [NSC16]. In Section 1.3, we assume $\mathbf{n}^- = -\mathbf{n}^+$ for the purpose of a simplified presentation of stabilisation methods.

for unit vector $\hat{\mathbf{z}}$ in the $\hat{z}$ direction.

H(curl) conforming. Similarly to H(div) conforming elements, H(curl) conforming ones can be created by ensuring the correct continuity at cell boundaries. In this case, since Stokes' theorem applies to surfaces and leads to line integrals, this applies to the fields' tangential component along edges (rather than faces) of three-dimensional cells. Three-dimensional box elements are therefore created via a tensor product of the form

$$\begin{aligned} H(\mathrm{curl})_k = (\hat{\mathbf{x}} DG_{k-1} \otimes CG_k \otimes CG_k) \oplus (CG_k \otimes \hat{\mathbf{y}} DG_{k-1} \otimes CG_k) \\ \oplus (CG_k \otimes CG_k \otimes \hat{\mathbf{z}} DG_{k-1}), \end{aligned} \tag{1.18}$$

which leads to function evaluation nodes of the fields' tangential components along edges (see Figure 1.2). In the two-dimensional case, the curl operator reduces to $\nabla^\perp = \mathbf{k} \times \nabla$, and can be interpreted in two ways, given the domain's outward unit vector $\mathbf{k}$. As an operation on a vector $\mathbf{v}$, it is given by $(\mathbf{k} \times \nabla) \cdot \mathbf{v}$, while as an operation on a scalar ψ it reads

$$\nabla^\perp \psi = (\mathbf{k} \times \nabla)\psi. \tag{1.19}$$

In what follows, we consider the latter case, where $H(\mathrm{curl})$ is given by a scalar space. Corresponding curl conforming spaces can be created by considering the vertical component of the three-dimensional tensor product spaces dotted with $\mathbf{k}$, which in fact leads to CG tensor product spaces as described at (1.12) above.

Given a set of finite elements, the corresponding finite element spaces are constructed using cell mappings between the reference cell and cells appearing in the domain's tessellation. Next to ensuring that the resulting spaces are indeed conformal, these mappings are also required to be consistent with the compatibility conditions between spaces, which are described below. While not directly relevant to the work presented in the chapters below, they form an important part of the construction of finite element spaces, and a further description of them can be found in Appendix A.

Compatibility

Considering the solution spaces of the weak form (1.9) - (1.10) for the linearised shallow water equations, we find that

$$H(\mathrm{div}, \Omega) \quad \xrightarrow{\nabla \cdot} \quad L^2(\Omega), \tag{1.20}$$

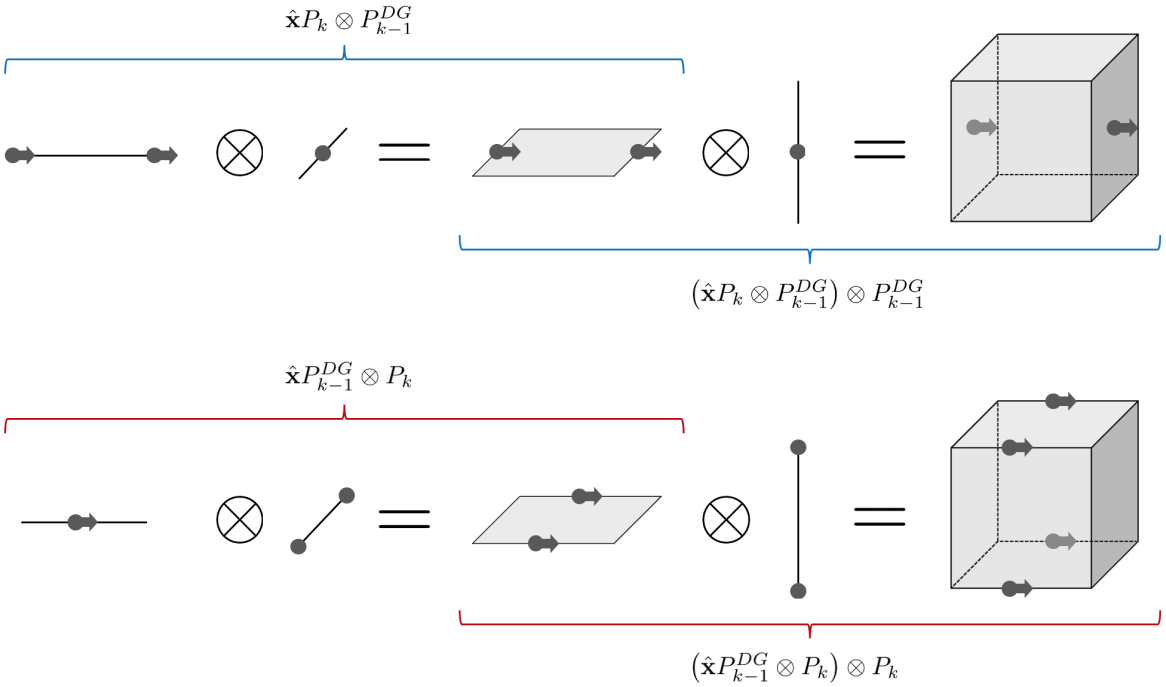


Figure 1.2: Tensor product construction for $\hat{\mathbf{x}}$-component of the direct sum used to create conforming shape function spaces for lowest degree $k = 1$. Top: H(div) element. Bottom: H(curl) element.

i.e. the former maps onto the latter via the divergence operator. The design of finite element spaces that fulfil Properties 2 and 3 mentioned in this section's introduction is aimed at maintaining a discrete version of (1.20), motivating Definitions 2 and 3 below [NSC16].

Definition 2. *Let Ω be a (topologically) two-dimensional domain without boundary. Further, let V_h^0, V_h^1, and V_h^2 be finite element subspaces of $H^1(\Omega)$, $H(\mathrm{div}, \Omega)$, and $L^2(\Omega)$, respectively. Then (V_h^0, V_h^1, V_h^2) are called compatible if*

i) $\nabla^\perp \eta \in V_h^1 \quad \forall \eta \in V_h^0$,

ii) $\nabla \cdot \mathbf{w} \in V_h^2 \quad \forall \mathbf{w} \in V_h^1$,

iii) there exist bounded projections π^i, $i = 0, 1, 2$, such that the following diagram commutes.

$$\begin{array}{ccccc} H^1(\Omega) & \xrightarrow{\nabla^\perp} & H(\mathrm{div};\Omega) & \xrightarrow{\nabla\cdot} & L^2(\Omega) \\ \downarrow{\scriptstyle \pi^0} & & \downarrow{\scriptstyle \pi^1} & & \downarrow{\scriptstyle \pi^2} \\ V_h^0(\Omega) & \xrightarrow{\nabla^\perp} & V_h^1(\Omega) & \xrightarrow{\nabla\cdot} & V_h^2(\Omega) \end{array}$$

An analogous set of compatible spaces exists for domains with boundaries, where $H^1(\Omega)$ and its discretised version are equipped with zero boundary conditions, while the two div-conforming spaces are equipped with zero normal component boundary conditions. In the chapters to follow, we will consider two-dimensional compatible finite element spaces based on the following sets of elements.

- $(P_{k+1},\ BDM_k,\ P_{k-1}^{DG})$ for triangular tessellations
- $(CG_k,\ H(\mathrm{div})_k,\ DG_{k-1})$ for square tessellations

Similarly, we can construct a set for three-dimensional domains, which contains four compatible spaces.

Definition 3. *Let Ω be a (topologically) three-dimensional domain without boundary. Further, let V_h^0, V_h^1, V_h^2, and V_h^3 be finite element subspaces of $H^1(\Omega)$, $H(\mathrm{curl},\Omega)$, $H(\mathrm{div},\Omega)$, and $L^2(\Omega)$, respectively. Then $(V_h^0,\ V_h^1,\ V_h^2,\ V_h^3)$ are called compatible if*

i) $\nabla\eta \in V_h^1 \quad \forall\eta \in V_h^0$,

ii) $\nabla \times \boldsymbol{\Sigma} \in V_h^2 \quad \forall\boldsymbol{\Sigma} \in V_h^1$,

iii) $\nabla \cdot \mathbf{w} \in V_h^3 \quad \forall\mathbf{w} \in V_h^2$,

iv) there exist bounded projections $\pi^i,\ i = 0,1,2,3$, such that the following diagram commutes.

$$\begin{array}{ccccccc} H^1(\Omega) & \xrightarrow{\nabla} & H(\mathrm{curl};\Omega) & \xrightarrow{\nabla\times} & H(\mathrm{div};\Omega) & \xrightarrow{\nabla\cdot} & L^2(\Omega) \\ \downarrow{\scriptstyle \pi^0} & & \downarrow{\scriptstyle \pi^1} & & \downarrow{\scriptstyle \pi^2} & & \downarrow{\scriptstyle \pi^3} \\ V_h^0(\Omega) & \xrightarrow{\nabla} & V_h^1(\Omega) & \xrightarrow{\nabla\times} & V_h^2(\Omega) & \xrightarrow{\nabla\cdot} & V_h^3(\Omega) \end{array}$$

In Chapter 4, we will consider three-dimensional compatible finite element spaces based on a set of tensor finite elements given by $(H(\mathrm{curl})_k,\ H(\mathrm{div})_k,\ DG_{k-1})$. Depictions of all compatible elements used thereafter are given in Figures 1.3 and 1.4.

Remark 4. *Next to the density related variable, the pressure related variable will also be considered as a function of the DG space. In particular, the choice of velocity and density related spaces determines the ratio of velocity and pressure degrees of freedom.*

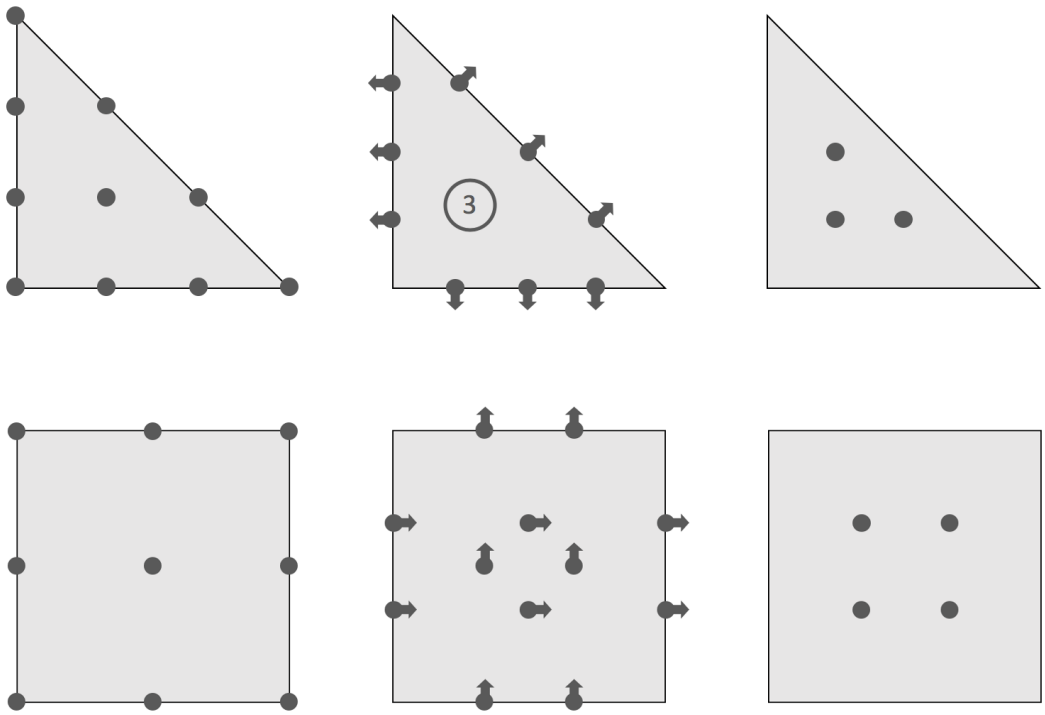


Figure 1.3: 2D finite elements used in compatible finite element setup, for degree $k = 2$. Top row: triangular elements P_{k+1}, BDM_k, P_{k-1}^{DG}. Bottom row: rectangular tensor product elements CG_k, $\mathrm{H(div)}_k$, DG_{k-1}. Dots and arrows correspond to degrees of freedom and degrees of freedom multiplied by a unit vector, respectively. Circled numbers indicate interior moments.

As mentioned in the motivation above, this in turn relates to the existence of certain computational wave modes. For the pair $(H(\mathrm{div})_k,\ DG_{k-1})$ *for square tessellations, we retrieve the correct ratio. For the pair* $\left(BDM_2,\ P_1^{DG}\right)$ *for triangular tessellations, the ratio can be shown to be equal to that of an Arakawa C discretisation on a hexagonal grid, leading to spurious Rossby wave modes that appear to have no damaging effect. For more details, see [CS12, TCD14].*

An important aspect of the compatible finite element spaces defined in the definitions above are their mimetic properties, i.e. they satisfy discrete vector calculus identities. This can be seen using commutativity, where e.g. in the two-dimensional case, given any $\eta \in V_h^0(\Omega)$, we can find $\psi \in H^1(\Omega)$ such that $\pi^0\psi = \eta$ and

$$\nabla \cdot \nabla^\perp(\pi^0\psi) = \pi^2(\nabla \cdot \nabla^\perp\psi) = \pi^2(0) = 0, \tag{1.21}$$

implying that the compatible finite element spaces satisfy the vector calculus identity

$$\nabla \cdot \nabla^\perp\eta = 0 \qquad\qquad \forall\eta \in V_h^0(\Omega). \tag{1.22}$$

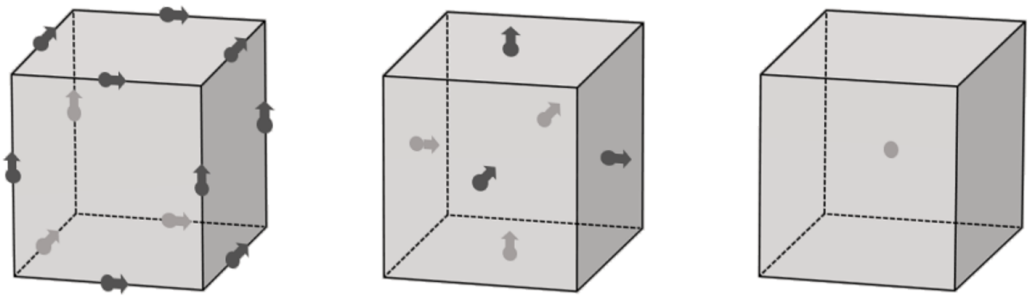


Figure 1.4: 3D tensor product finite element family $\mathsf{H(curl)}_k$, $\mathsf{H(div)}_k$, DG_{k-1} used in compatible finite element setup, for degree $k = 1$. Dots and arrows correspond to degrees of freedom and degrees of freedom multiplied by a unit vector, respectively.

1.1.3 Properties

Having defined the compatible finite element spaces, we are now in a position to space discretise the linearised weak shallow water equations (1.9) - (1.10). Next to presenting the discretisation, we will also review results related to mimetic properties. The review is largely based on [CS12] – the first paper to treat the method within the context of compatible mixed finite elements in numerical weather prediction – as well as the survey paper [NSC16]. Finally, we include a direct derivation of energy conservation, and discuss the derivation's extension to the nonlinear rotating shallow water equations.

Given a tessellation $\mathcal{T}$ of the domain Ω, the discretised inner product is given by

$$\langle v, w\rangle_h \coloneqq \sum_{K\in\mathcal{T}} \int_K v \cdot w \; dx, \tag{1.23}$$

where $(v,\ w)$ is a pair of scalar or vector fields, and each integral over K is computed via the reference element as described in Appendix A. For brevity, we drop subscripts h thereafter. Further, in view of the discretisations introduced below, we will refer to meshes rather than tessellations, noting that the former can be seen as an object containing information about cells, faces, edges, and vertices, while the latter is often simply understood as a partition of Ω. The space discretised linear shallow water equations are then given by

$$\langle \mathbf{w}, \mathbf{u}_t + f\mathbf{u}^\perp\rangle - \langle \nabla\cdot\mathbf{w}, gD\rangle = 0 \qquad \forall \mathbf{w} \in V_h^1(\Omega), \tag{1.24}$$

$$\langle \phi, D_t + H\nabla\cdot\mathbf{u}\rangle = 0 \qquad \forall \phi \in V_h^2(\Omega), \tag{1.25}$$

for prognostic variables $\mathbf{u} \in V_h^1(\Omega)$, $D \in V_h^2(\Omega)$, and test functions $\mathbf{w} \in V_h^1(\Omega)$, $\phi \in V_h^2(\Omega)$. Note that mass conservation follows by choosing a test function of the form $\phi = 1$, since then

$$\langle \phi, H\nabla \cdot \mathbf{u} \rangle = \sum_K \int_K \nabla \cdot (H\mathbf{u})\, dx = -\sum_K \int_{\partial K} H\mathbf{u} \cdot \mathbf{n}\, dx = 0, \tag{1.26}$$

recalling that H corresponds to the constant mean depth. The last equation follows since $\mathbf{u}$ has continuous normal components, implying that all cell boundary contributions cancel out (and we further assume $\mathbf{u} \cdot \mathbf{n} = 0$ on $\partial\Omega$).

Geostrophic balance

An atmospheric flow is in geostrophic balance if the pressure gradient and Coriolis terms sum to zero. In the case of the linear shallow water equations on a planar domain with the f-plane approximation (i.e. constant Coriolis parameter f [Val17]), we find that in the presence of such a balance, the strong equations (1.7) - (1.8) reduce to

$$f\mathbf{u}^\perp + g\nabla D = 0, \tag{1.27}$$

$$\mathbf{u}_t = 0, \tag{1.28}$$

$$D_t + H\nabla \cdot \mathbf{u} = 0. \tag{1.29}$$

Applying $\nabla^\perp$ to the balance equation (1.27), it follows that $\nabla \cdot \mathbf{u} = 0$, and therefore $D_t = 0$. In other words, in this case the geostrophically balanced state is steady. In the presence of nonlinear terms and a non-constant f, this state evolves as a slowly varying balanced state that characterises large scale weather systems. Maintaining the steady state property for the linearised equations after discretisation is crucial for an accurate representation of balanced large scale flow [CS12].

To demonstrate the existence of exact steady geostrophic states for the compatible finite element spaces introduced above, we start with a stream function $\psi \in V_h^0(\Omega)$. Given Definition 2, the resulting velocity field $\mathbf{u} = \nabla^\perp\psi$ is a function in the velocity space $V_h^1(\Omega)$. The discretised div-curl identity (1.22) then implies $\nabla \cdot \mathbf{u} = \nabla \cdot \nabla^\perp\psi = 0$. From the discretised continuity equation (1.25), we therefore find $D_t = 0$, and it remains to find a depth field D such that the Coriolis force and pressure gradient terms in the discretised velocity equation (1.24) are balanced. For this purpose, we define D such that

$$g\langle \phi, D \rangle = f\langle \phi, \psi \rangle \qquad\qquad \forall \phi \in V_h^2(\Omega), \tag{1.30}$$

i.e. the depth field is given by the projection of $(f/g)\psi$ into the discretised depth space $V_h^2(\Omega)$. Since $\nabla \cdot \mathbf{w}$ is a function of $V_h^2(\Omega)$ for any test function $\mathbf{w} \in V_h^1(\Omega)$, we then find from (1.30)

$$\langle \mathbf{w}, f\mathbf{u}^\perp \rangle - \langle \nabla \cdot \mathbf{w}, gD \rangle = \langle \mathbf{w}, f(\nabla^\perp \psi)^\perp \rangle - \langle \nabla \cdot \mathbf{w}, gD \rangle \tag{1.31}$$

$$= -\langle \mathbf{w}, f\nabla\psi \rangle - \langle \nabla \cdot \mathbf{w}, gD \rangle \tag{1.32}$$

$$= \langle \nabla \cdot \mathbf{w}, f\psi \rangle - \langle \nabla \cdot \mathbf{w}, gD \rangle = 0 \qquad \forall \mathbf{w} \in V_h^1(\Omega), \tag{1.33}$$

noting that in the Coriolis term we used integration by parts. The latter is performed cell-wise, and no facet integrals arise since all test functions $\mathbf{w}$ have continuous normal components and ψ is continuous. In the presence of boundaries, free-slip boundary conditions are assumed. Therefore, given a stream function ψ, the resulting velocity field together with a depth field of the form (1.30) lead to a geostrophically balanced state, which is steady in D (due to $\nabla \cdot \mathbf{u} = 0$) and $\mathbf{u}$ (inserting (1.33) into (1.24)).

Absence of pressure gradient induced vorticity

Taking the curl of the linearised momentum equation (1.7), we obtain an evolution equation of the form

$$\zeta_t + f\nabla \cdot \mathbf{u} = 0, \tag{1.34}$$

for vorticity $\zeta \coloneqq \nabla^\perp \cdot \mathbf{u}$, and constant f. In particular, for steady geostrophic states, where from the continuity equation (1.25) we have $\nabla \cdot \mathbf{u} = 0$, we find that vorticity is constant in time. In the compatible finite element setup, a discrete analogue to this can be derived by considering the vorticity as a function of $V_h^0(\Omega)$, defined weakly by

$$\langle \zeta, \eta \rangle = -\langle \nabla^\perp \eta, \mathbf{u} \rangle \qquad \forall \eta \in V_h^0(\Omega). \tag{1.35}$$

Differentiating the above with respect to time, we then find from the momentum equation (1.24),

$$\langle \eta, \zeta_t + f\nabla \cdot \mathbf{u} \rangle = \langle \nabla^\perp \eta, f\mathbf{u}^\perp \rangle - \langle \nabla \cdot \nabla^\perp \eta, gD \rangle + \langle \eta, f\nabla \cdot \mathbf{u} \rangle \tag{1.36}$$

$$= \langle \nabla\eta, f\mathbf{u} \rangle + \langle \eta, f\nabla \cdot \mathbf{u} \rangle \tag{1.37}$$

$$= -\langle \eta, f\nabla \cdot \mathbf{u} \rangle + \langle \eta, f\nabla \cdot \mathbf{u} \rangle = 0 \qquad \forall \eta \in V_h^0(\Omega) \tag{1.38}$$

noting that $\nabla^\perp \eta \in V_h^1(\Omega)$. Similarly to (1.33), we used integration by parts, and further also exploited the discrete div-curl identity (1.22). In particular, the latter ensures that

$$\langle \nabla \cdot \nabla^\perp \eta, gD \rangle = 0, \tag{1.39}$$

so that there is no unphysical source of gradient induced forcing in the discretised vorticity equation. Note that the div-curl identity will also apply in the same way for any other gradient related terms, such as the pressure gradient and geopotential gradient terms in the dry compressible Euler equations.

Energy conservation

Next to the existence of steady geostrophic modes and absence of pressure induced forcing terms in the vorticity equation, the compatible finite element framework also leads to energy conservation for the linearised shallow water equations. The system's total energy is given by

$$E = \frac{1}{2}\int_\Omega (H|\mathbf{u}|^2 + gD^2)dx, \tag{1.40}$$

i.e. the sum of kinetic and potential energy. For its rate of change in time, we then find directly from the discretised equations (1.24) and (1.25) with test functions $\mathbf{w} = H\mathbf{u}$ and $\phi = gD$, respectively,

$$\frac{dE}{dt} = \int_\Omega (H\mathbf{u}_t \cdot \mathbf{u} + gD_t D)dx \tag{1.41}$$

$$= \int_\Omega \Big(- fH\mathbf{u}\cdot\mathbf{u}^\perp + \big(\nabla\cdot(H\mathbf{u})\big)gD - H(\nabla\cdot\mathbf{u})gD \Big)dx = 0. \tag{1.42}$$

In order to motivate the Hamiltonian framework presented in Section 1.2 below, we next consider conservation of energy for the full non-linear set of rotating shallow water equations. For this purpose, we study their strong form, which is given by

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + f\mathbf{u}^\perp + g\nabla D = 0, \tag{1.43}$$

$$D_t + \nabla\cdot(\mathbf{u}D) = 0. \tag{1.44}$$

In this case, the total energy is given by

$$E = \frac{1}{2}\int_\Omega (D|\mathbf{u}|^2 + gD^2)dx, \tag{1.45}$$

and conservation of E for the strong form can be be computed by deriving evolution equations for the kinetic and potential energies from the momentum and continuity equations

[Val17]. For the point-wise potential energy $e_p = \frac{1}{2}gD^2$, we obtain from the continuity equation (1.44) multiplied by gD,

$$\frac{\partial e_p}{\partial t} + \nabla \cdot (\mathbf{u} e_p) + e_p \nabla \cdot \mathbf{u} = 0. \tag{1.46}$$

Similarly, from the momentum equation (1.43) multiplied by $\mathbf{u}D$, we obtain a point-wise kinetic energy evolution equation of the form

$$\frac{\partial e_k}{\partial t} + \nabla \cdot (\mathbf{u} e_k) + \mathbf{u} \cdot \nabla e_p = 0, \tag{1.47}$$

for kinetic energy $e_k = \frac{1}{2}D|\mathbf{u}|^2$, noting that we also used the continuity equation and the so-called vector invariant form

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{u}^\perp(\nabla^\perp \cdot \mathbf{u}) + \frac{1}{2}|\mathbf{u}^2|, \tag{1.48}$$

of the velocity advection term. Adding the two evolution equations together, we arrive at an energy equation of the form

$$\frac{\partial e}{\partial t} = \frac{\partial (e_k + e_p)}{\partial t} + \nabla \cdot \big(\mathbf{u}(e_k + 2e_p)\big) = 0, \tag{1.49}$$

for point-wise energy e. Finally, integrating over the domain and assuming no or free-slip boundaries, we find that $\mathrm{E} = \int_\Omega e \, dx$ is conserved, noting that the energy flux term $\mathbf{u}(e_k + 2e_p)$ vanishes after applying the divergence theorem.

While in the linear case conservation of energy is nearly trivial, this is far less so the case for the full rotating shallow water equations. The complexity can be expected to further increase in the discretised case, where only operations that are consistent with the discretisation at hand are allowed. This motivates a framework with a more explicit mechanism underlying energy conservation, as described in Section 1.2 below. Note that while the total energy in the latter section will be denoted by H (for Hamiltonian), here we instead used E to avoid confusion with the mean depth H.

1.1.4 Thermal field spaces

It remains to define a set of thermal field spaces used in conjunction with the compatible finite element spaces defined above. These will be needed for the potential temperature in the dry Euler equations, as well as the buoyancy in the thermal rotating shallow water equations. For the latter, we will set the space to be equal to the continuous space V_0^h.

It is also possible to choose the discontinuous space V_2^h instead, and for a discussion thereof in an energy conserving compatible context, we refer to [EDK19]. The choice of continuous space reflects our presentation of a new energy conserving stabilisation method for thermal field advection, as detailed in Chapter 3. For the dry Euler equations, we consider the continuous space V_0^h as well as the finite element equivalent of a Charney-Phillips finite difference grid, where the space's node locations are set to coincide with the velocity space nodes corresponding to the vertical wind [NSC16]. This choice of temperature space was recommended in a dispersion property study in [MBTC18] for mixed finite element methods in NWP when compared to a fully discontinuous or fully continuous finite element space, and will be used in the UK Met Office's next dynamical core. The resulting Charney-Phillips and triangular CG elements for the thermal field space in the case of the 2D Euler and thermal shallow water equations are given in Figure 1.5. The corresponding 3D Euler element is analogous to the 2D one, and is based on the vertical component of the 3D velocity element.

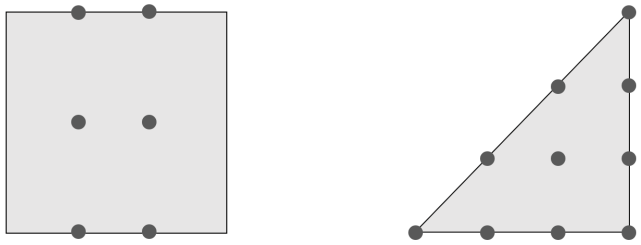


Figure 1.5: Reference elements for next to lowest order Charney-Phillips type finite element space and P_3. Dots denote degrees of freedom. The former is used for the Euler equations in a vertical slice of the atmosphere; the latter corresponds to a triangular mesh of the sphere used for the thermal shallow water equations.

1.2 Hamiltonian framework for energy conservation

Motivated by the discussion on energy conservation in Section 1.1.3, we next describe the Hamiltonian framework and use it for an alternative formulation of the rotating shallow water, thermal rotating shallow water, and dry compressible Euler equations. Many fluid dynamical equations can be formulated within such a framework [Mor82, Mor98], using the system's Hamiltonian H, i.e. the total amount of energy, and a Poisson bracket $\{.,.\}$. The latter is a form that maps from pairs of functionals of the prognostic variables to $\mathbb{R}$,

and satisfies for any functionals F, G, K [She90]

$$\begin{aligned} &\{\alpha F + \beta G, K\} = \alpha\{F, K\} + \beta\{G, K\} \quad \forall \alpha, \beta \in \mathbb{R} \\ &\{F, \alpha G + \beta K\} = \alpha\{F, G\} + \beta\{F, K\} \quad \forall \alpha, \beta \in \mathbb{R} \end{aligned} \qquad \text{(bilinearity)}, \tag{1.50}$$

$$\{F, G\} = -\{G, F\} \qquad \text{(antisymmetry)}, \tag{1.51}$$

$$\{F, \{G, K\}\} + \{G, \{K, F\}\} + \{K, \{F, G\}\} = 0 \qquad \text{(Jacobi identity)}. \tag{1.52}$$

The time evolution of any functional F of the prognostic variables is then given by the so-called Poisson system

$$\frac{dF}{dt} = \{F, H\}. \tag{1.53}$$

In the chapters below, we consider functionals of prognostic variables $\mathbf{u}$, ρ, and θ in a domain Ω, which denote the flow velocity, a density related field, and a thermal type field, respectively. Further, we will use the absolute vorticity $\boldsymbol{\omega}$, given by

$$\boldsymbol{\omega} = (\nabla \times \mathbf{u} + 2\boldsymbol{\Omega}), \tag{1.54}$$

for rotation vector $\boldsymbol{\Omega}$. Note that in the two-dimensional test cases considered in the numerical results sections of the chapters to follow, we will replace $\boldsymbol{\omega}$ by $(q\rho)\mathbf{k}$, for the 2D domain's outward unit vector $\mathbf{k}$ and scalar potential vorticity

$$q = (\nabla^{\perp} \cdot \mathbf{u} + f)/\rho, \tag{1.55}$$

recalling that $\nabla^{\perp}$ denotes the 2D curl operator $\mathbf{k} \times \nabla$, and f is given by the Coriolis parameter depending on the test case. This way, we can consider the discussion below both for 2D and 3D cases, without the need for two sets of notation.

In the chapters to follow, we will base our energy conserving, upwind stabilised formulations on a bracket as given in Definition 5 below, which we denote as the standard Poisson bracket or simply the *standard bracket*. It is given in terms of variational derivatives of functionals $F(\mathbf{u}, \rho, \theta)$, which are defined weakly by

$$\left\langle \frac{\delta F}{\delta \mathbf{u}}, \mathbf{w} \right\rangle := \lim_{\epsilon \to 0} \frac{1}{\epsilon} \big(F(\mathbf{u} + \epsilon \mathbf{w}, \rho, \theta) - F(\mathbf{u}, \rho, \theta) \big) \qquad \forall \mathbf{w} \in V(\Omega), \tag{1.56}$$

where as before, $\langle ., . \rangle$ denotes the L^2 inner product over Ω. $V(\Omega)$ is given by a suitable velocity space to be defined, and the variations in ρ and θ are defined analogously. In the continuous setting, we consider continuously differentiable spaces for V, which are then

discretised using suitable finite element spaces.

Definition 5. *For any functionals F, H of the prognostic variables $(\mathbf{u}, \rho, \theta)$, the* ***standard bracket*** *$\{F, H\} \in \mathbb{R}$ is given by*

$$\begin{aligned}\{F, H\} \coloneqq &- \left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla \theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \\ &- \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{1}{\rho} \frac{\delta F}{\delta \theta} \nabla \theta, \frac{\delta H}{\delta \mathbf{u}} \right\rangle .\end{aligned} \tag{1.57}$$

Note that this bracket, first introduced in [MG80], is indeed a Poisson bracket. It is clearly bilinear, and antisymmetry follows since the first term is antisymmetric in itself, while the second and fourth, as well as third and fifth terms form antisymmetric pairs, respectively. A direct proof for Jacobi's identity can be found e.g. in [Mor82], and a more straightforward derivation via the so-called Hamiltonian parcel mesh formulation can be found in [BO06]. For brevity, we exclude the proofs here, noting that we will find the energy conserving property of this framework to be purely a consequence of the bracket's antisymmetry. Jacobi's identity is associated with the system's so-called symplectic property, which in turn is related to conservation of area in phase space. Discretisations that preserve this property are aimed at an improved long term stability, and are commonly referred to as symplectic integrators; see e.g. [DR04] for a survey on such integrators and [Rei99] for a more rigorous analysis. Due to the above bracket's complexity, we do not aim to retain Jacobi's identity in the discretisations to follow and refer to e.g. [HLW06, LR04] for more details on this topic. Further, we note that this bracket is not unique, in the sense that the governing equations derived in this section may also be derived from different brackets. This includes brackets with functionals that depend on momentum $\mathbf{m} = \rho\mathbf{u}$ rather than velocity $\mathbf{u}$, and one such bracket is given in Appendix B. In particular, the bracket above is termed the *standard bracket* solely for the purpose of the chapters to follow, rather than from a general literature point of view.

The resulting fluid dynamical equations follow from their respective Hamiltonians and functionals F corresponding to weak forms of the prognostic variables. Specifically, we set $F = \langle \mathbf{u}, \mathbf{v} \rangle$, $F = \langle \rho, \psi \rangle$, and $F = \langle \theta, \chi \rangle$, respectively, with arbitrary test functions $\mathbf{v}, \psi, \chi$ in the corresponding spaces $V(\Omega)$. For $F = \langle \mathbf{u}, \mathbf{v} \rangle$, we have

$$\frac{\delta F}{\delta \mathbf{u}} = \mathbf{v}, \quad \frac{\delta F}{\delta \rho} = 0, \quad \frac{\delta F}{\delta \theta} = 0, \tag{1.58}$$

so that the Poisson bracket reduces to

$$\{F, H\} = -\left\langle \mathbf{v}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{v} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \mathbf{v} \right\rangle, \tag{1.59}$$

for any test function $\mathbf{v} \in V(\Omega)$. Using the Poisson system (1.53), we then obtain a momentum equation given by

$$\langle \mathbf{v}, \mathbf{u}_t \rangle = \frac{dF}{dt} = -\left\langle \mathbf{v}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{v} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \mathbf{v} \right\rangle. \tag{1.60}$$

Similarly, for ρ and θ, we obtain weak versions of the continuity and transport equations of the form

$$\langle \psi, \rho_t \rangle = -\left\langle \psi, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle, \tag{1.61}$$

$$\langle \chi, \theta_t \rangle = -\left\langle \chi, \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \cdot \nabla\theta \right\rangle, \tag{1.62}$$

for any test functions ψ, χ.

Given a system of equations of the form (1.60) - (1.62), conservation of energy then follows directly from the bracket framework. First, the Poisson system (1.53) with bracket (1.57) can be recovered from (1.60) - (1.62) for any functional $F(\mathbf{u}, \rho, \theta)$ via

$$\frac{dF}{dt} = \left\langle \frac{\delta F}{\delta \mathbf{u}}, \mathbf{u}_t \right\rangle + \left\langle \frac{\delta F}{\delta \rho}, \rho_t \right\rangle + \left\langle \frac{\delta F}{\delta \theta}, \theta_t \right\rangle = \{F, H\}. \tag{1.63}$$

In particular, this also holds for $F = H$, and noting the bracket's antisymmetry, we arrive at

$$\frac{dH}{dt} = \{H, H\} = -\{H, H\} = 0. \tag{1.64}$$

In what follows, we compute (1.60) - (1.62) for Hamiltonians corresponding to the rotating shallow water, thermal rotating shallow water, and dry compressible Euler equations. These sets of equations can be seen as a hierarchy, in the sense that their corresponding Hamiltonians grow increasingly complex: the rotating shallow water Hamiltonian does not contain an internal energy term, while the one for the thermal rotating shallow water equations does contain one in polynomial form. Finally, the Hamiltonian corresponding to the dry compressible Euler equations contains a non-polynomial internal energy term as specified by the ideal gas law. Given the Poisson bracket framework detailed above, we can then conclude directly that the resulting sets of equations conserve energy. Crucially, this remains true after the underlying spaces $V(\Omega)$ are discretised, and it further remains

true as long as the underlying discretised Poisson bracket is antisymmetric.

1.2.1 Rotating shallow water equations

The rotating shallow water equations have first been studied in the context of an energy conserving compatible finite element discretisation in [MC14], and a Poisson bracket equal to the standard one in Definition 5 (up to the thermal field θ related terms) can be found in the appendix therein. The Hamiltonian is given by

$$H(\mathbf{u},\rho) = \frac{1}{2}\int_\Omega \big(\rho|\mathbf{u}|^2 + g(\rho+b)^2\big)dx, \tag{1.65}$$

where in this context ρ corresponds to the fluid depth (and for consistency of notation with the Euler equations below, we keep ρ instead of using the more usual D or h), and there is no thermal field θ. Further, g and b denote the gravitational acceleration parameter and topographic height field, respectively. In view of the Poisson system (1.53), in the non-discretised case the variational derivatives are given by

$$\frac{\delta H}{\delta \mathbf{u}} = \rho\mathbf{u}, \qquad\qquad \frac{\delta H}{\delta \rho} = \frac{1}{2}|\mathbf{u}|^2 + g(\rho+b). \tag{1.66}$$

The usual form of the rotating shallow water equations then follows from (1.60) and (1.61). For the velocity equation, we have

$$\begin{aligned}
\langle \mathbf{v}, \mathbf{u}_t\rangle &= -\left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega}\times\frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla\cdot\frac{\delta F}{\delta \mathbf{u}}\right\rangle && (1.67)\\
&= -\langle \mathbf{v}, \boldsymbol{\omega}\times\mathbf{u}\rangle + \left\langle \frac{1}{2}|\mathbf{u}|^2 + g(\rho+b), \nabla\cdot\mathbf{v}\right\rangle && (1.68)\\
&= -\langle \mathbf{v}, (\nabla\times\mathbf{u} + 2\boldsymbol{\Omega})\times\mathbf{u}\rangle - \left\langle \frac{1}{2}\nabla|\mathbf{u}|^2 + \nabla(g\big(\rho+b\big)), \mathbf{v}\right\rangle && (1.69)\\
&= -\langle \mathbf{v}, (\mathbf{u}\cdot\nabla)\mathbf{u} + (2\boldsymbol{\Omega}\times\mathbf{u}) + g\nabla(\rho+b)\rangle\,. && (1.70)
\end{aligned}$$

where we have used integration by parts (with $\mathbf{v}\cdot\mathbf{n} = 0$ on $\partial\Omega$) and $(\mathbf{u}\cdot\nabla)\mathbf{u} = (\nabla\times\mathbf{u})\times\mathbf{u} + \frac{1}{2}\nabla|\mathbf{u}|^2$. Since this holds for any test function $\mathbf{v}$ in $C^1(\Omega)$, in the case of sufficiently smooth solutions we then arrive at the usual strong form of the momentum equation, given by

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + f\mathbf{u}^\perp + g\nabla(\rho+b) = 0. \tag{1.71}$$

Given the spherical context of the shallow water equations, we rewrote the Coriolis term as $f\mathbf{u}^\perp$, for Coriolis parameter $f = 2\Omega z/a$ and perpendicular $\mathbf{u}^\perp = \mathbf{k}\times\mathbf{u}$, where a, Ω,

z, and $\mathbf{k}$ correspond to the sphere's radius, rotation rate, rotational axis coordinate, and outward unit vertical vector, respectively. Similarly, we obtain the continuity equation

$$\rho_t + \nabla \cdot (\rho \mathbf{u}) = 0, \tag{1.72}$$

where we used $\frac{\delta H}{\delta \mathbf{u}} = \rho \mathbf{u}$ in (1.61), which holds for any test function ψ.

For completeness, we note that the linear rotating shallow water equations used in the previous section can be derived analogously, with a Hamiltonian of the form (1.40) and a bracket that excludes the nonlinear velocity advection term (i.e. the first term in (1.57)).

1.2.2 Thermal rotating shallow water equations

The thermal shallow water equations have originally been introduced in [Rip93], and describe a vertically averaged form of the rotating stratified hydrostatic Euler equations. Here we follow the formulation as given in [EDK19]. The Hamiltonian is given by

$$H(\mathbf{u}, \rho, \theta) = \int_\Omega \big(\frac{\rho}{2}|\mathbf{u}|^2 + \rho\theta(\frac{\rho}{2} + b)\big) dx, \tag{1.73}$$

where θ corresponds to the buoyancy. Note that for $\theta = g$, we recover the previous shallow water Hamiltonian (up to a constant of the form $\int_\Omega g b^2 \, dx$). This time, the variational derivatives are given by

$$\frac{\delta H}{\delta \mathbf{u}} = \rho \mathbf{u}, \qquad \frac{\delta H}{\delta \rho} = \frac{1}{2}|\mathbf{u}|^2 + \theta(\rho + b), \tag{1.74}$$

$$\frac{\delta H}{\delta \theta} = \rho(\frac{\rho}{2} + b), \tag{1.75}$$

and we obtain a weak momentum equation of the form

$$\langle \mathbf{v}, \mathbf{u}_t \rangle = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \tag{1.76}$$

$$= -\langle \mathbf{v}, \boldsymbol{\omega} \times \mathbf{u} \rangle + \left\langle \frac{1}{2}|\mathbf{u}|^2 + \theta(\rho + b), \nabla \cdot \mathbf{v} \right\rangle + \left\langle (\frac{\rho}{2} + b)\nabla\theta, \mathbf{v} \right\rangle \tag{1.77}$$

$$= -\langle \mathbf{v}, (\nabla \times \mathbf{u} + 2\boldsymbol{\Omega}) \times \mathbf{u} \rangle - \left\langle \frac{1}{2}\nabla|\mathbf{u}|^2 + \nabla\big(\theta(\rho + b)\big), \mathbf{v} \right\rangle + \left\langle (\frac{\rho}{2} + b)\nabla\theta, \mathbf{v} \right\rangle \tag{1.78}$$

$$= -\left\langle \mathbf{v}, (\mathbf{u} \cdot \nabla)\mathbf{u} + (2\boldsymbol{\Omega} \times \mathbf{u}) + \theta\nabla(\rho + b) + \frac{\rho}{2}\nabla\theta \right\rangle. \tag{1.79}$$

In the non-discretised case, this leads to

$$\mathbf{u}_t + (\mathbf{u}\cdot\nabla)\mathbf{u} + f\mathbf{u}^\perp + \theta\nabla(\rho + b) + \frac{\rho}{2}\nabla\theta = 0, \tag{1.80}$$

which, as expected, reduces to the shallow water version (1.71) if $\theta = g$. Similarly, we obtain the continuity and temperature transport equations

$$\rho_t + \nabla\cdot(\rho\mathbf{u}) = 0, \tag{1.81}$$

$$\theta_t + \mathbf{u}\cdot\nabla\theta = 0, \tag{1.82}$$

where we additionally used $\frac{\delta H}{\delta \mathbf{u}} = \rho\mathbf{u}$ in (1.62), which holds for any test function χ.

1.2.3 Euler equations

A Poisson bracket based formulation of the dry compressible Euler equations can be found e.g. in [Gas13, LP20]. Note that the Poisson bracket and the skew symmetric operator presented in these papers correspond to a different Poisson bracket to the one considered here, relying on the use of a different set of underlying fields (with mass weighted potential temperature Θ). In our case, the Hamiltonian is given by

$$H(\mathbf{u},\rho,\theta) = \int_\Omega \left(\frac{\rho}{2}|\mathbf{u}|^2 + g\rho z + c_v\rho\theta\pi\right) dx, \tag{1.83}$$

for wind velocity $\mathbf{u}$, air density ρ, and potential temperature θ. z and c_v denote the vertical coordinate and specific heat of air at constant volume, respectively. Further, π denotes the Exner pressure, which is given by the ideal gas law as

$$\pi^{\frac{1-\kappa}{\kappa}} = \frac{R}{p_0}\rho\theta, \tag{1.84}$$

for reference pressure p_0, ideal gas constant R, and non-dimensional parameter $\kappa = R/c_p$, where $c_p = R + c_v$ denotes the specific heat at constant pressure. Proceeding as before, the variational derivatives are given by

$$\frac{\delta H}{\delta \mathbf{u}} = \rho\mathbf{u}, \qquad \frac{\delta H}{\delta \rho} = \frac{1}{2}|\mathbf{u}|^2 + gz + c_p\theta\pi, \tag{1.85}$$

$$\frac{\delta H}{\delta \theta} = c_p\rho\pi. \tag{1.86}$$

Note that to derive the pressure related expression for the variation in θ, we used (1.84) and expanded the pressure term according to

$$\pi(\rho, \theta + \epsilon\chi) = \left(\frac{R}{p_0}\rho\right)^{\frac{\kappa}{1-\kappa}} (\theta + \epsilon\chi)^{\frac{\kappa}{1-\kappa}} \tag{1.87}$$

$$= \left(\frac{R}{p_0}\rho\right)^{\frac{\kappa}{1-\kappa}} \left(\theta^{\frac{\kappa}{1-\kappa}} + \frac{\kappa}{1-\kappa}\theta^{\frac{\kappa}{1-\kappa}-1}\epsilon\chi + \mathcal{O}(\epsilon^2)\right) \tag{1.88}$$

$$= \pi(\rho, \theta) + \frac{R}{c_v}\frac{\pi(\rho, \theta)}{\theta}\epsilon\chi + \mathcal{O}(\epsilon^2), \tag{1.89}$$

for test function χ of the temperature space, and where we used $\frac{\kappa}{1-\kappa} = \frac{R}{c_v}$. The limit term in the variational derivative (1.56) can then be evaluated according to

$$\lim_{\epsilon\to 0} \frac{1}{\epsilon}\big(c_v\rho(\theta + \epsilon\chi)\pi(\rho, \theta + \epsilon\chi) - c_v\rho\theta\pi(\rho, \theta)\big) \tag{1.90}$$

$$= \lim_{\epsilon\to 0} \frac{1}{\epsilon}\left(c_v\rho(\theta + \epsilon\chi)\left(\pi(\rho, \theta) + \frac{R}{c_v}\frac{\pi(\rho, \theta)}{\theta}\epsilon\chi + \mathcal{O}(\epsilon^2)\right) - c_v\rho\theta\pi(\rho, \theta)\right) \tag{1.91}$$

$$= \lim_{\epsilon\to 0} \left(c_v\left(1 + \frac{R}{c_v}\right)\rho\pi(\rho, \theta)\chi + \mathcal{O}(\epsilon)\right) \tag{1.92}$$

$$= c_p\rho\pi(\rho, \theta)\chi. \tag{1.93}$$

The variation in ρ in (1.85) follows an analogous argument. As before, the Euler equations then follow from (1.60) - (1.62), with a velocity equation of the form

$$\langle \mathbf{v}, \mathbf{u}_t\rangle = -\left\langle \mathbf{v}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{v}\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \mathbf{v}\right\rangle \tag{1.94}$$

$$= -\langle \mathbf{v}, \boldsymbol{\omega} \times \mathbf{u}\rangle + \left\langle \frac{1}{2}|\mathbf{u}|^2 + gz + c_p\theta\pi, \nabla \cdot \mathbf{v}\right\rangle + \langle c_p\pi\nabla\theta, \mathbf{v}\rangle \tag{1.95}$$

$$= -\langle \mathbf{v}, (\nabla \times \mathbf{u} + 2\boldsymbol{\Omega}) \times \mathbf{u}\rangle - \left\langle \frac{1}{2}\nabla|\mathbf{u}|^2 + g\hat{\mathbf{z}} + c_p\nabla(\theta\pi), \mathbf{v}\right\rangle + \langle c_p\pi\nabla\theta, \mathbf{v}\rangle \tag{1.96}$$

$$= -\left\langle \mathbf{v}, (\mathbf{u} \cdot \nabla)\mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u} + g\hat{\mathbf{z}} + c_p\theta\nabla\pi\right\rangle, \tag{1.97}$$

for vertical unit vector $\hat{\mathbf{z}}$ (not to be confused with the domain's outward unit vector $\mathbf{k}$). In strong form, this leads to

$$\mathbf{u}_t + (\mathbf{u} \cdot \nabla)\mathbf{u} + 2\boldsymbol{\Omega} \times \mathbf{u} + g\hat{\mathbf{z}} + c_p\theta\nabla\pi = 0. \tag{1.98}$$

Note that the pressure gradient is given in terms of potential temperature θ and Exner pressure π. To arrive at its familiar form with density ρ and pressure p, we note that the

latter is given by the more common version of the ideal gas law (1.84) of the form

$$p = \rho R T, \tag{1.99}$$

where the temperature T is related to π and θ by $T = \theta\pi$. Expanding $c_p = R + c_v$ and using the chain rule for $\nabla\pi$, we then find

$$\begin{aligned} c_p\theta\nabla\pi =& R\theta\nabla\pi + c_v\theta\nabla\left(\frac{R}{p_0}\rho\theta\right)^{\frac{\kappa}{1-\kappa}} = R\theta\nabla\pi + c_v\theta\frac{\kappa}{1-\kappa}\left(\frac{R}{p_0}\rho\theta\right)^{\frac{\kappa}{1-\kappa}-1}\nabla\left(\frac{R}{p_0}\rho\theta\right) && (1.100)\\ =& R\theta\nabla\pi + c_v\theta\frac{R}{c_v}\frac{\pi}{\frac{R}{p_0}\rho\theta}\nabla\left(\frac{R}{p_0}\rho\theta\right) && (1.101)\\ =& R\theta\nabla\pi + R\frac{\pi}{\rho}\nabla(\rho\theta) = \frac{1}{\rho}\nabla(R\rho\theta\pi) = \frac{1}{\rho}\nabla p. && (1.102) \end{aligned}$$

Similarly to the thermal shallow water case, equations for ρ and θ follow from (1.61) and (1.62), and since the variational derivative $\frac{\delta H}{\delta \mathbf{u}}$ is the same here, the resulting equations are given as before by

$$\rho_t + \nabla\cdot(\rho\mathbf{u}) = 0, \tag{1.103}$$

$$\theta_t + \mathbf{u}\cdot\nabla\theta = 0. \tag{1.104}$$

Remark 6. *Next to deriving sets of governing equations, the Poisson system* (1.53) *can also readily be used to determine transfers between different types of energy. For instance, the Hamiltonian* (1.83) *corresponding to the compressible Euler equations can be expressed as a sum $H = K + P + I$, for kinetic, potential, and internal total energies*

$$K = \frac{1}{2}\int_\Omega \rho|\mathbf{u}|^2\,dx, \qquad P = \int_\Omega g\rho z\,dx, \qquad I = \int_\Omega c_v\rho\theta\pi\,dx. \tag{1.105}$$

Setting the functional F in the Poisson system equal to K, P, and I, respectively, we find

$$\begin{aligned} \frac{dK}{dt} =& +\langle\nabla\cdot\mathbf{F}, gz\rangle + c_p\Big(\langle\nabla\cdot\mathbf{F}, \theta\pi\rangle + \langle T/\rho, \mathbf{F}\cdot\nabla\theta\rangle\Big) && (1.106)\\ \frac{dP}{dt} =& -\langle\nabla\cdot\mathbf{F}, gz\rangle && (1.107)\\ \frac{dI}{dt} =& -c_p\Big(\langle\nabla\cdot\mathbf{F}, \theta\pi\rangle + \langle T/\rho, \mathbf{F}\cdot\nabla\theta\rangle\Big), && (1.108) \end{aligned}$$

for $\mathbf{F} = \frac{\delta H}{\delta \mathbf{u}} = \rho\mathbf{u}$ and $T = \frac{\delta H}{\delta\theta}/c_p = \rho\pi$. Note that given the discussion in the section below, we find that the above derivation of energy transfers via the Poisson system is

also applicable after discretisation. The Poisson bracket framework therefore in principle provides an approach to examine whether the discretisation correctly represents energy transfers. In particular, as long as the bracket's qualitative structure of antisymmetric pairs is left intact, there should not be unbalanced transfers, or unphysical ones such as potential to internal energy.

1.2.4 Discretisation

Note that we obtained the governing equations (1.60) - (1.62) by choosing appropriate functionals F in the Poisson system (1.53). In view of energy conservation, we discretise at the level of the Poisson system and use discrete versions $\mathbb{V}(\Omega)$ of the continuous spaces $V(\Omega)$ to obtain the resulting discretised governing equations. In particular, if the space-discretised system is still based on an antisymmetric bracket, the resulting set of equations will also still be energy conserving. Note that in the discretised case, the Hamiltonian variations are given by projections into the chosen finite element spaces. For instance, for the rotating shallow water equations on a domain tessellated by a triangular mesh, we can pick the space triple $(\mathbb{V}_q,\ \mathbb{V}_u,\ \mathbb{V}_\rho) = (P_3,\ BDM_2,\ P_1^{DG})$ for vorticity, velocity, and density, and the Hamiltonian derivatives are then given by

$$\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_u}(\rho \mathbf{u}) =: \mathbf{F}, \qquad \frac{\delta H}{\delta \rho} = P_{\mathbb{V}_\rho}\left(\frac{1}{2}|\mathbf{u}|^2 + g(\rho + b)\right), \tag{1.109}$$

for projections $P_{\mathbb{V}_u}$ and $P_{\mathbb{V}_\rho}$ into $\mathbb{V}_u$, $\mathbb{V}_\rho$, respectively. Using the standard bracket (1.57) and functionals F analogous to the continuous case (with test functions from the finite element spaces), this then leads to an energy conserving space discretisation of the form [MC14]

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = -\left\langle \mathbf{w}, q\mathbf{F}^\perp \right\rangle + \left\langle \frac{1}{2}|\mathbf{u}|^2 + g(\rho + b), \nabla \cdot \mathbf{w} \right\rangle \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{1.110}$$

$$\langle \phi, \rho_t \rangle = -\langle \phi, \nabla \cdot \mathbf{F} \rangle \qquad \forall \phi \in \mathbb{V}_\rho, \tag{1.111}$$

recalling $\mathbf{F}^\perp = \mathbf{k} \times \mathbf{F}$. Note that the projection for the variation in ρ is not needed in the momentum equation since the compatible setup guarantees $\nabla \cdot \mathbf{w} \in \mathbb{V}_\rho$. Further, note that since $\mathbf{F} \in \mathbb{V}_u$, we have $\nabla \cdot \mathbf{F} \in \mathbb{V}_D$, and from the continuity equation (1.111) we therefore find that D_t and $-\nabla \cdot \mathbf{F}$ are equivalent in $\mathbb{V}_D \subset L^2(\Omega)$, in the sense that $\|D_t - (-\nabla \cdot \mathbf{F})\|_2 = 0$. The discrete vorticity $q \in \mathbb{V}_q$ appearing in the bracket (and therefore in the governing equations) is specified via a discrete version of the diagnostic

relation (1.55), given by

$$\langle \eta, \rho q \rangle = -\langle \nabla^{\perp} \eta, \mathbf{u} \rangle + \langle \eta, f \rangle \qquad \forall \eta \in \mathbb{V}_q, \tag{1.112}$$

for 2D scalar curl as given in (1.19). Note that the derivation of the Poisson system given the prognostic equations, as shown in (1.63), still holds in the discretised case. Since the bracket is identical to the standard one up to discretised terms (and hence antisymmetric), the discretisation is indeed still energy conserving. However, for the discretised bracket and all other discretised ones introduced thereafter, it is not clear if the Jacobi identity still holds. For this reason, we will refer to the discretised antisymmetric brackets considered here as almost Poisson brackets, which is the usual terminology for such brackets in the literature.

Equations (1.60) - (1.62) derived from the Poisson system can also be time-discretised in an energy conserving way, using a so-called Poisson integrator as introduced in [CH11]. For the prognostic fields $\mathbf{z} = (\mathbf{u}, \rho, \theta)$, they can be reformulated to

$$\dot{\mathbf{z}}(t) = J(\mathbf{z}) \frac{\delta H}{\delta \mathbf{z}}, \tag{1.113}$$

for $t \geq 0$, and where the skew symmetric transformation J is related to the almost Poisson bracket via

$$\{F, H\} = \left\langle \frac{\delta F}{\delta \mathbf{z}}, J(\mathbf{z}) \frac{\delta H}{\delta \mathbf{z}} \right\rangle. \tag{1.114}$$

Further, we set

$$\frac{\delta H}{\delta \mathbf{z}} = \Big(\frac{\delta H}{\delta \mathbf{u}}, \frac{\delta H}{\delta \rho}, \frac{\delta H}{\delta \theta} \Big)^T. \tag{1.115}$$

The discretisation is then given by

$$\mathbf{z}^{n+1} = \mathbf{z}^n + \Delta t J\Big(\frac{\mathbf{z}^{n+1} + \mathbf{z}^n}{2} \Big) \Big(\overline{\frac{\delta H}{\delta \mathbf{u}}}, \overline{\frac{\delta H}{\delta \rho}}, \overline{\frac{\delta H}{\delta \theta}} \Big)^T, \tag{1.116}$$

for time step Δt, and prognostic fields $\mathbf{z}^k$ at time $k\Delta t$, for $k \in \mathbb{Z}_{\geq 0}$. The time averaged Hamiltonians are given by

$$\overline{\frac{\delta H}{\delta \mathbf{u}}} \coloneqq \int_0^1 \frac{\delta}{\delta \mathbf{u}} H\big(\mathbf{z}^n + s(\mathbf{z}^{n+1} - \mathbf{z}^n) \big) ds, \tag{1.117}$$

and similarly for the variations in ρ and θ.

To illustrate that the above Poisson integrator does indeed conserve energy, we first note

that (1.116) follows from discretising the time derivative $\dot{\mathbf{z}}(t)$ according to

$$\dot{\mathbf{z}}(t_n + \tau\Delta t) = J\left(\mathbf{z}\left(t_n + \frac{\Delta t}{2}\right)\right)\int_0^1 \frac{\delta}{\delta \mathbf{z}} H\big(\mathbf{z}(t_n + s\Delta t)\big) ds, \tag{1.118}$$

for $t_n \in \{n\Delta t \mid n \in \mathbb{Z}_{\geq 0}\}$ and $\tau \in [0,1]$. Note that the right-hand side is given by a polynomial approximation of the system in the interval $[t_n, t_{n+1}]$, and corresponds to a discretisation that is of second order in time. Higher order ones can be constructed in a similar fashion by using approximations of a higher polynomial order; see [CH11]. Integrating (1.118) over $[t_n, t_{n+1}]$, we then obtain (1.116), noting that the equality

$$\mathbf{z}(t_n + s\Delta t) = \mathbf{z}^n + s(\mathbf{z}^{n+1} - \mathbf{z}^n) \tag{1.119}$$

follows directly by considering (1.118) integrated over $[t_n, s\,t_{n+1}]$ as well as $[t_n, t_{n+1}]$. Finally, conservation of energy can be demonstrated via the fundamental theorem of calculus, which yields

$$H(\mathbf{z}^{n+1}) - H(\mathbf{z}^n) = \Delta t \int_0^1 \frac{\delta}{\delta \mathbf{z}} H\big(\mathbf{z}(t_n + \tau\Delta t)\big)\, \dot{\mathbf{z}}(t_n + \tau\Delta t)\, d\tau \tag{1.120}$$

$$= \Delta t \int_0^1 \frac{\delta}{\delta \mathbf{z}} H\big(\mathbf{z}(t_n + \tau\Delta t)\big)\, J\left(\mathbf{z}\left(t_n + \frac{\Delta t}{2}\right)\right)\int_0^1 \frac{\delta}{\delta \mathbf{z}} H\big(\mathbf{z}(t_n + s\Delta t)\big) ds\, d\tau \tag{1.121}$$

$$= \Delta t\, \overline{\frac{\delta H}{\delta \mathbf{z}}}^T J\left(\mathbf{z}\left(t_n + \frac{\Delta t}{2}\right)\right)\overline{\frac{\delta H}{\delta \mathbf{z}}} = 0, \tag{1.122}$$

noting that J is skew symmetric, and where we wrote $\overline{\frac{\delta H}{\delta \mathbf{z}}}$ for $\frac{\delta H}{\delta \mathbf{z}}$ with time averaged variations of the form (1.117).

The time averaged expression (1.117) can be integrated exactly for the Hamiltonian variation in $\mathbf{u}$, leading to

$$\overline{\frac{\delta H}{\delta \mathbf{u}}} = \frac{1}{3} P_{\mathbb{V}_u}\big(\rho^n \mathbf{u}^n + \frac{1}{2}\rho^n \mathbf{u}^{n+1} + \frac{1}{2}\rho^{n+1}\mathbf{u}^n + \rho^{n+1}\mathbf{u}^{n+1}\big). \tag{1.123}$$

The same holds true for the Hamiltonian variations in ρ and θ for the rotating shallow water and thermal rotating shallow water equations. A fully energy conserving scheme

for the rotating shallow water equations is therefore given by

$$\left\langle \mathbf{v}, \mathbf{u}^{n+1} - \mathbf{u}^n \right\rangle = \Delta t \left(- \left\langle \mathbf{w}, \bar{q} \overline{\frac{\delta H}{\delta \mathbf{u}}}^{\perp} \right\rangle + \left\langle \overline{\frac{\delta H}{\delta \rho}}, \nabla \cdot \mathbf{w} \right\rangle \right) \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{1.124}$$

$$\left\langle \phi, \rho^{n+1} - \rho^n \right\rangle = -\Delta t \left\langle \phi, \nabla \cdot \overline{\frac{\delta H}{\delta \mathbf{u}}} \right\rangle \qquad \forall \phi \in \mathbb{V}_\rho, \tag{1.125}$$

where q is part of the transformation J and therefore time-discretised as $\bar{q} = (q^n + q^{n+1})/2$. Note that in general, the nonlinear systems of equations that result from this discretisation procedure depend on the underlying choice of Hamiltonian in the Poisson system. Solution strategies for these nonlinear systems are presented in the numerical results sections of Chapters 2 and 3. They are given by iterative methods, and the number of iterations will generally have an impact on the degree of energy conservation. Further, for the Euler Hamiltonian, we find that the internal energy is a non-polynomial function in ρ and θ, thus requiring an approximate integration of (1.117) for the variations in ρ and θ. For the numerical results presented in Chapters 3 and 4, a fourth order Gaussian quadrature is used. While the resulting scheme is no longer strictly energy conserving, we found that higher order quadratures did not reduce the energy error. This is believed to be due to other more significant energy error sources, such as linear solver tolerance and the number of nonlinear iterations.

Finally, while it may seem natural to choose a bracket that is a straight discretised version of the standard one, the resulting equations may lead to undesirable features with respect to stability, as illustrated in Appendix C. The main contribution of this thesis and its underlying published work concerns the design of discretised almost Poisson brackets that are antisymmetric and lead to sets of governing equations that include stabilisation methods.

1.3 Upwind stabilisation methods

Next to Coriolis and gradient related terms, which combine to the typical large scale balances in the atmosphere, the governing equations presented in the previous section also contain terms related to advection. It is well-known that a straight discretisation of these terms may lead to spurious oscillations (see e.g. [QV08]), and in this section we review methods designed to reduce them. This includes methods for the compatible finite element and thermal field spaces introduced above, which will then be incorporated into the Hamiltonian framework in the chapters below. Further, we will briefly also mention

other stabilisation techniques, which address topics such as simplicity, other aspects of the forecasting model, preventing unphysical (i.e. negative) values, other physical interpretations, and conceptual ideas related to the finite element method. An illustration of the oscillations' origin, as well as their mitigation by means of the classical DG upwind and SUPG methods, is given in Appendix C.

1.3.1 Classical DG upwind stabilisation

In this section, we introduce an upwind stabilisation method for discontinuous Galerkin spaces. Note that according to the compatible finite element framework, we will use a DG discretisation for the continuity equation, and will therefore present the stabilisation method by means of the latter. It is given by

$$\langle \phi, \rho_t \rangle + \langle \phi, \nabla \cdot (\mathbf{u}\rho) \rangle = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \tag{1.126}$$

for the discontinuous Galerkin finite element space $\mathbb{V}_\rho$ used for the density related field.

Considering a given cell K of the mesh, we obtain from integration by parts

$$\int_K \phi \nabla \cdot (\mathbf{u}\rho) dx = -\int_K \nabla\phi \cdot (\mathbf{u}\rho) dx + \int_{\partial K} \phi\rho\mathbf{u} \cdot \mathbf{n}\, dS, \tag{1.127}$$

for cell boundary ∂K and its corresponding unit outward normal vector $\mathbf{n}$. However, since ρ is an element of a DG space, it is multivalued on ∂K, and to resolve this, the density related field on ∂K is defined by its value on the boundary's upwind side [Kuz10]. Summing over all cells, we obtain the classical DG upwind stabilisation method, given by

$$\langle \phi, \rho_t \rangle = \langle \nabla\phi, \mathbf{u}\rho \rangle - \int_\Gamma [\![\phi\mathbf{u}]\!] \tilde{\rho}\, dS \qquad \forall \phi \in \mathbb{V}_\rho, \tag{1.128}$$

where the last integral is over all interior mesh facets, noting that we assume free-slip boundary conditions for $\mathbf{u}$. The jump operator $[\![.]\!]$ (for vectors $\mathbf{v}$ and scalars ψ, respectively) and upwind value $\tilde{\rho}$ are defined by

$$\begin{aligned} [\![\mathbf{v}]\!] &= \mathbf{v}^+ \cdot \mathbf{n}^+ + \mathbf{v}^- \cdot \mathbf{n}^-, \\ [\![\psi]\!] &= \psi^+ - \psi^-, \end{aligned} \qquad \tilde{\rho} = \begin{cases} \rho^+ & \text{if } \mathbf{u} \cdot \mathbf{n}^+ < 0 \\ \rho^- & \text{otherwise,} \end{cases} \tag{1.129}$$

where the two sides of each mesh facet are arbitrarily denoted by + and - (and hence $\mathbf{n}^+ = -\mathbf{n}^-$). When considering (1.128) as a weak form before discretisation, with smooth test functions ϕ, we find that the facet term in (1.128) vanishes. Therefore, the incorporation

of upwinding is still consistent, in the sense that a solution to the underlying differential equation also solves the discretised one. Further, like the non-stabilised flux based version (1.111), the upwind formulation is mass conserving, since choosing $\phi = 1$ leads to $\nabla\phi = 0$ and $[\![\phi\mathbf{u}]\!] = [\![\mathbf{u}]\!] = 0$. The latter follows from our choice of velocity function space, noting that $\mathbf{u}$ has continuous normal components across facets.

1.3.2 Lie derivative type upwinding

The standard bracket (1.57) is based on the vector invariant form of the momentum equations, with an advection term of the form

$$(\mathbf{u}\cdot\nabla)\mathbf{u} = (\nabla\times\mathbf{u})\times\mathbf{u} + \frac{1}{2}\nabla|\mathbf{u}|^2. \tag{1.130}$$

This form is similar to the so-called Lie derivative, which can be seen as a generalisation of the scalar advection operator (for details see Appendix B and e.g. [MR13]). For the velocity field, it reads

$$\pounds_{\mathbf{v}}(\mathbf{u}) = (\nabla\times\mathbf{u})\times\mathbf{v} + \nabla(\mathbf{v}\cdot\mathbf{u}), \tag{1.131}$$

for advecting velocity $\mathbf{v}$ and advected velocity $\mathbf{u}$. Note that in (1.130) we have $\mathbf{v} = \mathbf{u}$, which may, however, not be the case in the energy conserving discretisations presented in the chapters below. In [NC17b], an energy conserving discretisation was introduced for perfect incompressible fluids, relying on a discretisation framework of (1.131) for H(div) conforming spaces. One possible discretisation in this framework is given by the upwind form

$$\pounds^h_{\mathbf{v}}(\mathbf{u};\ \mathbf{w}) = \langle\nabla\times(\mathbf{v}\times\mathbf{w}),\mathbf{u}\rangle - \int_\Gamma \{\{\mathbf{n}\times(\mathbf{v}\times\mathbf{w})\}\}\cdot\tilde{\mathbf{u}}\,dS - \langle\nabla\cdot\mathbf{w},\mathbf{v}\cdot\mathbf{u}\rangle\,, \tag{1.132}$$

for any test function $\mathbf{w}$ of the velocity space $\mathbb{V}_u$, and where $\tilde{\mathbf{u}}$ is defined analogously to $\tilde{\rho}$ above, and further

$$\{\{\mathbf{n}\times\mathbf{a}\}\} = \mathbf{n}^+\times\mathbf{a}^+ + \mathbf{n}^-\times\mathbf{a}^-, \tag{1.133}$$

for any given vector field $\mathbf{a}$. Similarly to the DG upwind case, (1.132) is obtained from its continuous version (1.131) by multiplying by a test function, integrating by parts, and defining the resulting facet integral accordingly. Note that specifying $\tilde{\mathbf{u}}$ is sound in the sense that while the normal component of $\mathbf{u}$ is continuous across facets, this need not be the case for the field $\mathbf{u}$ itself. Further details on this type of upwind stabilisation in terms of its motivation, convergence and energy spectrum properties can be found in [NC17a, NC17b].

1.3.3 SUPG stabilisation

Having defined two upwind stabilisation methods that exploit the discontinuity across element boundaries, it remains to specify such a method for CG spaces. Here, we consider the Streamline Upwind Petrov Galerkin method, and to illustrate it we start with the thermal field transport equation in its continuous form, given by

$$\theta_t + \mathbf{u} \cdot \nabla\theta = 0. \tag{1.134}$$

In a finite element setting, we would space discretise the transport equation using its weak form

$$\langle \gamma, \theta_t + \mathbf{u} \cdot \nabla\theta \rangle = 0 \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{1.135}$$

for thermal field space $\mathbb{V}_\theta$ as discussed in Section 1.1.4. To avoid spurious oscillations, diffusion can be added in the direction of the flow [Kuz10], which is achieved by replacing the test functions by

$$\gamma \to \gamma + \tau\mathbf{u} \cdot \nabla\gamma =: \gamma_{\mathbf{u}}, \tag{1.136}$$

for a suitable coefficient τ (see Appendix C.3.1). We then arrive at

$$\langle \gamma_{\mathbf{u}}, \theta_t + \mathbf{u} \cdot \nabla\theta \rangle = 0 \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{1.137}$$

and this form of upwinding is commonly referred to as the Streamline Upwind Petrov Galerkin (SUPG) method [Kuz10], first introduced in [BH82]. The form is a Petrov Galerkin method, i.e. one where test and trial functions originate from distinct spaces, since the modified test functions (1.136) need no longer be elements of $\mathbb{V}_\theta$. Setting $\gamma = \theta \in \mathbb{V}_\theta$, the SUPG equation (1.137) leads to

$$\frac{1}{2}\frac{d}{dt}\|\theta\|_2^2 = -\langle \theta, \mathbf{u} \cdot \nabla\theta \rangle - \|\sqrt{\tau}\mathbf{u} \cdot \nabla\theta\|_2^2 - \langle \tau\mathbf{u} \cdot \nabla\theta, \theta_t \rangle, \tag{1.138}$$

where the indefinite SUPG term (i.e. the last term) is kept small by a suitable choice of τ. In particular, we then find that the negative definite term $-\|\sqrt{\tau}\mathbf{u} \cdot \nabla\theta\|_2^2$ acts as diffusion along streamlines of θ. Note that since all test functions were modified to include an upwind term, the resulting weak equation (1.137) is still clearly consistent, in the sense that an exact solution to the continuous equation (1.134) also solves the discretised weak equation.

The stabilised advection equation (1.137) can be used for fully continuous spaces $\mathbb{V}_\theta$, as is the case e.g. for our choice of buoyancy space for the thermal shallow water equations. For spaces that are not fully continuous, such as the Charney-Phillips type finite element space, further care needs to be taken for the discontinuous components. In our case, we follow [NSC16], and apply the classical DG upwind method in the horizontal (i.e. discontinuous) direction, and introduce the SUPG method in the vertical direction only. To this end, we first integrate the weak form (1.135) by parts and arrive at

$$\langle \gamma, \theta_t \rangle - \langle \nabla \cdot (\mathbf{u}\gamma), \theta \rangle + \int_{\Gamma_v} [\![\mathbf{u}\gamma]\!] \tilde{\theta} \, dS = 0 \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{1.139}$$

where Γ_v denotes the set of all vertical interior facets of the underlying mesh, noting that $\mathbf{u}\gamma$ is continuous across horizontal facets. The jump operator and upwind value $\tilde{\theta}$ are defined as for the classical DG method above. Before incorporating the upwind modification (1.136) to the test functions in (1.139), we need to integrate by parts again to avoid applying the differential operator ∇ to the upwinded part $\tau \mathbf{u} \cdot \nabla \gamma$, as the double differentiation may not be well-defined for $\gamma \in \mathbb{V}_\theta$. Further, we restrict the SUPG method to the vertical direction by using modified test functions of the form

$$\gamma \to \gamma + \tau(\hat{\mathbf{z}} \cdot \mathbf{u})(\hat{\mathbf{z}} \cdot \nabla \gamma) =: \gamma_{(\hat{\mathbf{z}} \cdot \mathbf{u})\hat{\mathbf{z}}}, \tag{1.140}$$

for vertical unit vector $\hat{\mathbf{z}}$. The resulting, fully upwinded advection equation is then given by

$$\left\langle \gamma_{(\hat{\mathbf{z}} \cdot \mathbf{u})\hat{\mathbf{z}}}, \theta_t \right\rangle + \left\langle \gamma_{(\hat{\mathbf{z}} \cdot \mathbf{u})\hat{\mathbf{z}}}, \mathbf{u} \cdot \nabla \theta \right\rangle + \int_{\Gamma_v} \left([\![\mathbf{u}\gamma_{(\hat{\mathbf{z}} \cdot \mathbf{u})\hat{\mathbf{z}}}]\!] \tilde{\theta} - [\![\mathbf{u}\gamma_{(\hat{\mathbf{z}} \cdot \mathbf{u})\hat{\mathbf{z}}} \theta]\!] \right) dS = 0 \quad \forall \gamma \in \mathbb{V}_\theta, \tag{1.141}$$

noting that the second integration by parts leads to another facet integral term. In Chapter 3, we will incorporate the stabilised advection methods (1.137) and (1.141) in the standard bracket (1.57). To simplify notation, we write the upwinded test functions (1.136) and (1.140) as well as advection terms in (1.137) and (1.141) as operators given in the definition below, where CG and CP denote continuous Galerkin and Charney-Phillips, respectively.

Definition 7. *The SUPG contribution to $\gamma \in \mathbb{V}_\theta$ is given by*

$$\tilde{\tilde{\gamma}}_{\mathbf{u}} := \begin{cases} \tau \mathbf{u} \cdot \nabla \gamma & \text{CG type } \mathbb{V}_\theta, \\ \tau(\hat{\mathbf{z}} \cdot \mathbf{u})(\hat{\mathbf{z}} \cdot \nabla \gamma) & \text{CP type } \mathbb{V}_\theta. \end{cases} \tag{1.142}$$

Further, for an upwinded thermal field function

$$\gamma_{\mathbf{u}} \coloneqq \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \tag{1.143}$$

the discrete thermal field transport operator is given by

$$L^{\theta}_{\mathbf{v}}(\theta;\gamma_{\mathbf{u}}) = \begin{cases} \left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \mathbf{v}\cdot\nabla\theta \right\rangle & \textit{CG type } \mathbb{V}_\theta, \\ \left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \mathbf{v}\cdot\nabla\theta \right\rangle + \int_{\Gamma_v} \left([\![\mathbf{v}(\gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}})]\!] \tilde{\theta} - [\![\mathbf{v}(\gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}) \theta]\!] \right) dS & \textit{CP type } \mathbb{V}_\theta. \end{cases} \tag{1.144}$$

Using this definition, we can rewrite the advection equations (1.137) and (1.141) as a single equation of the form

$$\left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle + L^{\theta}_{\mathbf{u}}(\theta;\gamma_{\mathbf{u}}) = 0 \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{1.145}$$

Finally, we note that the lowered $\mathbf{u}$ at L corresponds to the advecting velocity, while the lowered $\mathbf{u}$ at γ corresponds to the velocity used for upwinding. As mentioned before, depending on the time discretisation, we will find that these may be distinct in the almost Poisson bracket setup.

1.3.4 Other types of stabilisation

In the chapters to follow, we will combine the standard bracket (1.57) with the aforementioned advection stabilisation methods. Before moving on to the first main chapter, which concerns the incorporation of classical DG upwinding, we briefly mention other stabilisation methods that are commonly used in atmosphere models, following [MKM+16].

Viscosity based stabilisation

There are many reasons to include dissipation in an atmospheric model, and next to stabilising the numerical model, this is done e.g. to eliminate noise originating from initialisation or grid-scale forcing from the physics parameterisations, to prevent the accumulation of potential enstrophy at the smallest grid scales, and to model sub-grid scale turbulence [JW11]. The most common mean of dissipation in current weather forecast models is given by the addition of an artificial diffusion term of the form [MKM+16]

$$\int_\Omega (-1)^{\alpha+1} p_h \nabla^\alpha \cdot (\nu_{2\alpha} \nabla q_h)\, dx, \tag{1.146}$$

for test function p_h, prognostic field q_h, and diffusivity coefficient matrix $\nu_{2\alpha}$. The choice of $\alpha \geq 1$ determines the degree of hyperviscosity, with higher order frequencies damped more strongly as α increases. This method is robust and simple to implement, and a choice of $\alpha > 1$ is often motivated in the sense that this leads small oscillations caused by numerical error to be damped, while leaving large physical modes untouched. On the other hand, a common criticism to this method is that if (1.146) does not tend to zero as the mesh size tends to zero, the resulting numerical method does not solve for the original equation set [MKM+16].

There also exist alternative approaches to add viscosity in a scale selective way. In the context of the so-called spectral method, which in numerical weather prediction relies on spherical harmonics together with Fourier transforms, this can be done by damping higher order modes (i.e. "filtering") to control high-wavenumber oscillations [GH01]. As the resolution is increased by including a larger number of Fourier modes, the damping effect decreases as the latter only applies to the highest modes. For this reason, this method is also referred to as Spectral Vanishing Viscosity (SVV). The concept can also be adapted to spectral element methods, where polynomial filtering is applied (for details, see [KS06]).

In our context, an argument can be made against artificial diffusion by means of (1.146) for the purpose of stabilising the numerical model. Depending on the choice of mesh and spaces, the compatible finite element method may avoid the computational modes that have to be controlled via artificial diffusion, and the advection terms may be controlled e.g. by the consistent stabilisation methods detailed above. Further, an alternative, vorticity equation based treatment for the accumulation of potential enstrophy will be discussed in Chapters 2 and 4.

Physics based stabilisation

As detailed in Appendix C.3, stabilisation methods can be seen as sub-grid models to treat unresolved scales. The behaviour in these scales can be studied from a physics point of view in the context of Large Eddy Simulation (LES), where a separation between resolved and unresolved scales is assumed for the continuous equations. This separation is achieved by a spatial filter [MKM+16]

$$\bar{\mathbf{u}} = \int_\Omega G_\Delta(\mathbf{x} - \hat{\mathbf{x}})\mathbf{u}(\hat{\mathbf{x}})d\hat{\mathbf{x}}, \tag{1.147}$$

for filtering function G_Δ with filter width Δ, leading to a set of governing equations that contain resolved and unresolved terms. For instance, setting $\tilde{\mathbf{u}} = \overline{\rho\mathbf{u}}/\bar{\rho}$, the momentum equation is then given by

$$(\bar{\rho}\tilde{\mathbf{u}})_t + \nabla\cdot(\bar{\rho}\tilde{\mathbf{u}}\otimes\tilde{\mathbf{u}}) + \nabla\bar{p} = \nabla\cdot\boldsymbol{\tau} - \bar{\rho}g, \tag{1.148}$$

and $\nabla\cdot\boldsymbol{\tau}$ represents the unresolved scale, for turbulent stress tensor

$$\boldsymbol{\tau} = \bar{\rho}\left(\widetilde{\mathbf{u}\otimes\mathbf{u}} - \tilde{\mathbf{u}}\otimes\tilde{\mathbf{u}}\right). \tag{1.149}$$

The unresolved terms are then replaced by physics based sub-grid models, which may include a dissipative and hence stabilising effect. Next to this effect, the methods are intended to capture the sub-grid scale dynamics and feed it into the numerical model. One classical such approach is given by the Smagorinsky sub-filter model [Sma63], which is based on assumptions related to energy transfer and dissipation. Another, more recent approach was presented in [MNG15] for a spectral element method based discretisation, where the turbulent stress tensor is approximated by

$$\boldsymbol{\tau} \approx \frac{\mu}{2}\left(\nabla\tilde{\mathbf{u}} + \nabla\tilde{\mathbf{u}}^T\right), \tag{1.150}$$

for dynamic viscosity parameter μ defined according to cell-wise residuals of the filtered governing equations (akin to (C.31)). Note that given this setup, $\nabla\cdot\boldsymbol{\tau}$ acts as dissipation in regions where the solution is under-resolved, i.e. where the cell-wise residual is large. For comparison purposes, we will consider this method in Remark 25 in Chapter 3.

Positivity preservation

While the aforementioned stabilisation methods may lead to a reduction of spurious oscillations, they do not necessarily preserve monotonicity, i.e. entirely prevent the generation of such oscillations. This is problematic in cases where fields must not exceed a certain threshold, which is the case in particular for tracer quantities such as moisture. Note that the latter may take small values in the atmosphere, but not negative ones. A common strategy to tackle this problem is the use of limiters, i.e. post-processing filters that constrain functions to stay within certain bounds [Kuz10]. Such limiters can be combined both with CG and DG methods [MKM+16], and a vertex based limiter as described in [Kuz10] has recently been considered for the compatible finite element method in the context of the moist compressible Euler equations in [BGS+19].

Other finite element based types of stabilisation

While classical DG upwinding and SUPG stabilisation are specific to the finite element method, the stabilisation methods considered in Section 1.3.4 can also be applied to other underlying types of space discretisation. Here, we briefly mention other well-known stabilisation methods specific to the finite element method. To begin with, note that the SUPG formulation (1.137) can be rewritten as

$$\langle \gamma, \theta_t + \mathbf{u} \cdot \nabla\theta \rangle = - \langle \tau \mathbf{u} \cdot \nabla\gamma, \theta_{res} \rangle \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{1.151}$$

where the temperature equation's residual is given by

$$\theta_{res} = \theta_t + \mathbf{u} \cdot \nabla\theta. \tag{1.152}$$

For more general evolution equations of the form

$$\theta_t + \mathcal{L}(\theta) = 0, \tag{1.153}$$

where the dynamical operator $\mathcal{L}$ may e.g. also contain a diffusion term $-\nabla \cdot (\nu \nabla\theta)$ (for diffusion parameter ν), the stabilised weak form (1.151) can then also be generalised to

$$\langle \gamma, \theta_t + \mathcal{L}(\theta) \rangle = - \langle \tau \mathcal{P}(\gamma), \theta_{res} \rangle \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{1.154}$$

and the choice of $\mathcal{P}(\gamma)$ determines the method at hand. For $\mathcal{P}(\gamma) = \mathbf{u} \cdot \nabla\gamma$, we recover the SUPG method, while for $\mathcal{P} = \mathcal{L}$ we obtain a so-called Galerkin Least Squares (GLS) method. Further, for $\mathcal{P} = -\mathcal{L}^*$, i.e. the negative of the adjoint operator of $\mathcal{L}$, we recover a sub-grid scale (SGS) method. Note that depending on the form of $\mathcal{L}$, boundary conditions, and $\mathbf{u}$, these two generalisations may reduce to the SUPG method. For more details, see [Kuz10, MNG15] and the references therein.

Another approach, which is distinct from the above Petrov Galerkin type framework, is given by edge based stabilisation [BH04] (also called "continuous interior penalty method" [Kuz10]). The stabilisation follows by penalising gradient jumps across edges in 2D and faces in 3D according to

$$\int_\Gamma \kappa h_e^2 [\![\mathbf{n} \cdot \nabla\gamma]\!] [\![\mathbf{n} \cdot \nabla\theta]\!] \, dS, \tag{1.155}$$

for a tuning parameter κ and edge size h_e. Setting $\gamma = \theta$, we then obtain a term of the form

$$\int_\Gamma \kappa h_e^2 [\![\mathbf{n} \cdot \nabla\theta]\!]^2 \, dS, \tag{1.156}$$

which, like the streamline upwind term in (1.138), acts to reduce $\|\theta\|_2^2$. Note that adding (1.155) to the discretised weak form is consistent with smooth exact solution to the strong equations, since in this case $[\![\mathbf{n} \cdot \nabla\theta]\!] = 0$ and therefore (1.155) vanishes.

2

Density stabilisation

In this chapter, we introduce the incorporation of classical DG upwinding (1.128) into the standard bracket (1.57), and present numerical results by means of the rotating shallow water equations. Since upwind stabilisation for the thermal field will be presented in the next chapter, we consider a simplified version of the standard bracket here, which does not contain thermal field terms:

$$\begin{aligned}\{F,H\} = &-\left\langle \frac{\delta F}{\delta \mathbf{u}}, q\frac{\delta H}{\delta \mathbf{u}}^{\perp}\right\rangle + \left\langle \frac{\delta H}{\delta D}, \nabla\cdot\frac{\delta F}{\delta \mathbf{u}}\right\rangle \\ &- \left\langle \frac{\delta F}{\delta D}, \nabla\cdot\frac{\delta H}{\delta \mathbf{u}}\right\rangle,\end{aligned} \tag{2.1}$$

where in light of the shallow water equations based numerical results we also use the depth field D as the density related field, and the 2D version (1.55) of the vorticity term. Without upwind stabilisation, we recall that the momentum and continuity equations resulting from the Poisson system (1.53) are given as in (1.110) - (1.111) by

$$\begin{aligned}\langle \mathbf{v}, \mathbf{u}_t\rangle &= -\left\langle \mathbf{w}, q\mathbf{F}^{\perp}\right\rangle + \left\langle \frac{1}{2}|\mathbf{u}|^2 + g(D+b), \nabla\cdot\mathbf{w}\right\rangle && \forall \mathbf{w}\in\mathbb{V}_u,\\ \langle \phi, D_t\rangle &= -\langle \phi, \nabla\cdot\mathbf{F}\rangle && \forall \phi\in\mathbb{V}_D,\end{aligned} \tag{2.2}$$

for velocity and depth finite element spaces $\mathbb{V}_u \subset H(\mathrm{div};\Omega)$, $\mathbb{V}_D \subset L^2(\Omega)$, and Hamiltonian variations

$$\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_u}(D\mathbf{u}) =: \mathbf{F}, \qquad \frac{\delta H}{\delta D} = P_{\mathbb{V}_D}\left(\frac{1}{2}|\mathbf{u}|^2 + g(D+b)\right). \tag{2.3}$$

In order to include DG upwinding in the continuity equation of (2.2), we first note that the advection term is derived within the Poisson system from the last term in the simplified standard bracket (2.1), i.e. for a functional $F = \langle \phi, D \rangle$, we obtain

$$-\left\langle \frac{\delta F}{\delta D}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle \quad \rightarrow \quad -\langle \phi, \nabla \cdot \mathbf{F} \rangle. \tag{2.4}$$

To arrive at a DG upwind formulation of the advection term, we need to adjust the according bracket term. To maintain energy conservation, we then simply need to reestablish antisymmetry in the bracket, that is we also need to adjust the corresponding originally antisymmetric term (second term in (2.1)) accordingly. In Section 2.1, we present this mechanism for the continuity equation. The resulting methodology can be equally applied to the Lie derivate advection method (1.132), and in order to arrive at a fully upwinded fully energy conserving formulation for the shallow water equations, we also describe the latter in Section 2.1.2. In Section (2.2) we explain the treatment of the resulting system of nonlinear equations and present numerical results that confirm energy conservation and an improved field development if upwind stabilisation is included. Finally, in Section (2.3) we summarise the chapter.

2.1 DG upwind stabilisation in the discretised bracket

To incorporate classical DG upwinding (1.128) in an energy conserving discretisation, we first note that in the continuity equation within the energy conserving framework, the depth and velocity fields are combined to a flux field $\mathbf{F}$. On the other hand, the two fields appear separately in the upwind discretisation's facet integral term. Therefore, to arrive at the classical formulation, we introduce an additional operator to avoid the occurrence of projections due to the discrete Hamiltonian variations (1.109). In particular, we introduce an operator to recover our velocity $\mathbf{u}$ from $\mathbf{F}$, arising due to the Hamiltonian variation with respect to $\mathbf{u}$.

Definition 8. *Given a function* $\mathbf{m}$ *in the velocity finite element space* $\mathbb{V}_u$ *and a function* D *in the depth finite element space* $\mathbb{V}_D$*, the velocity recovery operator is given by*

$$\mathbb{U}(D, \mathbf{m}) \colon \mathbb{V}_D \times \mathbb{V}_u \longrightarrow \mathbb{V}_u \quad \textit{such that} \quad \langle D\mathbf{v}, \mathbb{U} \rangle = \langle \mathbf{v}, \mathbf{m} \rangle \quad \forall \mathbf{v} \in \mathbb{V}_u. \tag{2.5}$$

Note that in the continuous sense, $\mathbb{U}(D, \mathbf{m})$ corresponds to division of $\mathbf{m}$ by D. Since in our shallow water setting $D > 0$, we find that $\mathbb{U}$ is well-defined and further that if $\mathbf{m}$ is

any function in $\mathbb{V}_u$ of flux form $\mathbf{F} = P_{\mathbb{V}_u}(D\mathbf{u})$, then

$$\mathbb{U}(D, \mathbf{F}) = \mathbf{u}, \tag{2.6}$$

point-wise.

Using $\mathbb{U}$, we are in a position to alter bracket (2.1) by introducing upwinding terms for D, and arrive at a bracket of the form

$$\begin{aligned}\{F, H\} = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, q\frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle - \left\langle D\, \mathbb{U}\left(D, \frac{\delta F}{\delta \mathbf{u}}\right), \nabla \frac{\delta H}{\delta D} \right\rangle + \int_\Gamma \left[\!\left[\frac{\delta H}{\delta D} \mathbb{U}\left(D, \frac{\delta F}{\delta \mathbf{u}}\right) \right]\!\right] \tilde{D}\, dS \\ + \left\langle D\, \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right), \nabla \frac{\delta F}{\delta D} \right\rangle - \int_\Gamma \left[\!\left[\frac{\delta F}{\delta D} \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right) \right]\!\right] \tilde{D}\, dS,\end{aligned} \tag{2.7}$$

noting that we also had to introduce upwinding for the corresponding antisymmetric term of the momentum equation (second term in 2.1) to maintain antisymmetry. Further, note that (2.7) corresponds to the standard upwinding formulation (1.128), provided that (2.6) holds. To see this, we follow the standard Poisson bracket procedure, choosing $F = \langle D, \phi \rangle$ (for test function $\phi \in \mathbb{V}_D$) to find

$$\frac{\delta F}{\delta D} = \phi, \quad \frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_u}(D\mathbf{u}). \tag{2.8}$$

The flux recovery operator follows from its definition, i.e.

$$\left\langle D\mathbf{v}, \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right) \right\rangle = \left\langle \mathbf{v}, \frac{\delta H}{\delta \mathbf{u}} \right\rangle = \langle \mathbf{v}, P_{\mathbb{V}_u}(D\mathbf{u}) \rangle = \langle \mathbf{v}, D\mathbf{u} \rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u \tag{2.9}$$

$$\Rightarrow \quad \left\langle D\mathbf{v}, \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right) \right\rangle = \langle D\mathbf{v}, \mathbf{u} \rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u \tag{2.10}$$

$$\Rightarrow \quad \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right) = \mathbf{u}. \tag{2.11}$$

Using $\frac{\delta F}{\delta D}$ from (2.8) and $\mathbb{U}\big(D, \frac{\delta H}{\delta \mathbf{u}}\big) = \mathbf{u}$ in a Poisson system with bracket (2.7) then leads to

$$\langle \phi, D_t \rangle = \frac{dF}{dt} = \{F, H\} = \left\langle D\, \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right), \nabla \frac{\delta F}{\delta D} \right\rangle - \int_\Gamma \left[\!\left[\frac{\delta F}{\delta D} \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right) \right]\!\right] \tilde{D}\, dS \tag{2.12}$$

$$= \langle D\mathbf{u}, \nabla \phi \rangle - \int_\Gamma [\![\phi \mathbf{u}]\!] \tilde{D}\, dS, \tag{2.13}$$

i.e. the standard form (1.128). Note that this equality holds up to the definition of the flux recovery operator, which also depends on the choice of time discretisation, as described

in Section (2.2.1).

2.1.1 Flux dependent formulation

An alternative way of deriving an energy conserving, upwind stabilised formulation for the continuity equation is given by retaining the flux $\mathbf{F}$, and using a modified version of the standard upwind form (1.128). One possible such alternative form of the facet integral is given by

$$\int_\Gamma [\![\phi \mathbf{F}/D]\!] \tilde{D}\, dS. \tag{2.14}$$

However, we find that (2.14) is not mass conserving since $D \in \mathbb{V}_D \subset L^2(\Omega)$ may not be continuous across facets (and hence $[\![\mathbf{F}/D]\!] \neq 0$). To avoid this, we first note that for velocity spaces with continuous normal components across facets (such as the $H(\mathrm{div})$ spaces appearing in the compatible framework), the jump term in the upwind facet integral can be rewritten as

$$\begin{aligned}
[\![\phi \mathbf{u}]\!] =& \phi^+ \mathbf{u}^+ \cdot \mathbf{n}^+ + \phi^- \mathbf{u}^- \cdot \mathbf{n}^- && (2.15)\\
=& \phi^+ \mathbf{u}^+ \cdot \mathbf{n}^+ - \phi^- \mathbf{u}^+ \cdot \mathbf{n}^+ && (2.16)\\
=& (\phi^+ - \phi^-) \mathbf{u}^+ \cdot \mathbf{n}^+ && (2.17)\\
=& [\![\phi]\!] \mathbf{u}^+ \cdot \mathbf{n}^+. && (2.18)
\end{aligned}$$

Written in this form, we find that approximations to $\mathbf{u}$ will not affect mass conservation (as choosing $\phi = 1$ yields $[\![\phi]\!] = 0$). Setting

$$\mathbf{u}^+ \cdot \mathbf{n}^+ \approx \frac{1}{2}(\mathbf{F}^+/D^+ + \mathbf{F}^-/D^-) \cdot \mathbf{n}^+ = \mathrm{avg}(\mathbf{F}/D), \tag{2.19}$$

we obtain an alternative upwind formulation given by

$$\langle \phi, D_t \rangle = \langle \nabla \phi, \mathbf{F} \rangle - \int_\Gamma [\![\phi]\!] \mathrm{avg}(\mathbf{F}/D) \tilde{D}\, dS \qquad \forall \phi \in \mathbb{V}_D. \tag{2.20}$$

This form of upwind stabilisation can be retrieved from an almost Poisson bracket without the need of a velocity recovery operator, which is given by

$$\{F,H\} = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, q\frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle - \left\langle \frac{\delta F}{\delta \mathbf{u}}, \nabla\frac{\delta H}{\delta D} \right\rangle + \int_{\Gamma} \left[\!\left[\frac{\delta H}{\delta D} \right]\!\right] \text{avg}\left(\frac{\delta F}{\delta \mathbf{u}}/D \right) \tilde{D}\, dS \tag{2.21}$$

$$+ \left\langle \frac{\delta H}{\delta \mathbf{u}}, \nabla\frac{\delta F}{\delta D} \right\rangle - \int_{\Gamma} \left[\!\left[\frac{\delta F}{\delta D} \right]\!\right] \text{avg}\left(\frac{\delta H}{\delta \mathbf{u}}/D \right) \tilde{D}\, dS. \tag{2.22}$$

We find that since $\mathbf{F}$ is in the velocity space and thus also has continuous normal components, this approximation depends on the difference of D across each facet's side. This can be problematic whenever steep gradients occur, as will be discussed in the numerical results section below.

2.1.2 Velocity upwinding

It is also possible to apply the velocity recover operator $\mathbb{U}$ for the Lie derivative type stabilisation (1.132) of the advection term as reviewed in the previous chapter. In two dimensions, the continuous and discretised Lie derivatives are given by

$$\pounds_{\mathbf{v}}(\mathbf{u}) = (\nabla^{\perp}\cdot\mathbf{u})\mathbf{v}^{\perp} + \nabla(\mathbf{v}\cdot\mathbf{u}), \tag{2.23}$$

$$\pounds^{h}_{\mathbf{v}}(\mathbf{u};\ \mathbf{w}) = \left\langle \nabla^{\perp}(\mathbf{v}\cdot\mathbf{w}^{\perp}), \mathbf{u} \right\rangle - \int_{\Gamma} [\![\mathbf{v}\cdot\mathbf{w}]\!]\mathbf{n}^{\perp}\cdot\tilde{\mathbf{u}}\, dS - \left\langle \nabla\cdot\mathbf{w}, \mathbf{v}\cdot\mathbf{u} \right\rangle. \tag{2.24}$$

To this end, we first note that the weak gradient term (last term in (2.24)) appears in the Hamiltonian framework within the Hamiltonian variation in D. Second, for the curl related terms, we note that in non-discretised form, the bracket term corresponding to velocity advection reads

$$\left\langle \frac{\delta F}{\delta \mathbf{u}}, q\frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle = \left\langle \frac{\delta F}{\delta \mathbf{u}}, \left(\frac{\nabla^{\perp}\cdot\mathbf{u} + f}{D} \right) \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle \tag{2.25}$$

$$= \left\langle \frac{\delta F}{\delta \mathbf{u}}, (\nabla^{\perp}\cdot\mathbf{u}) \left(\frac{\delta H}{\delta \mathbf{u}}/D \right)^{\perp} \right\rangle + \left\langle \frac{\delta F}{\delta \mathbf{u}}, f \left(\frac{\delta H}{\delta \mathbf{u}}/D \right)^{\perp} \right\rangle. \tag{2.26}$$

In particular, the first term in (2.26) is analogous to the continuous Lie derivative with an advecting velocity of the form $\frac{\delta H}{\delta \mathbf{u}}/D$, suggesting a Lie derivative stabilisation within

the bracket of the form

$$\left\langle \frac{\delta F}{\delta \mathbf{u}}, q\frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle \to -\left\langle \nabla^{\perp}\left(\frac{\delta F}{\delta \mathbf{u}}\cdot\left(\frac{\delta H}{\delta \mathbf{u}}/D\right)^{\perp}\right), \mathbf{u}\right\rangle + \int_{\Gamma}\left[\!\left[\frac{\delta F}{\delta \mathbf{u}}\cdot\left(\frac{\delta H}{\delta \mathbf{u}}/D\right)^{\perp}\right]\!\right] \mathbf{n}^{\perp}\cdot\tilde{\mathbf{u}}\, dS$$

$$+\left\langle \frac{\delta F}{\delta \mathbf{u}}, f\left(\frac{\delta H}{\delta \mathbf{u}}/D\right)^{\perp}\right\rangle, \tag{2.27}$$

and we find that this form of velocity advection is still energy conserving, as all terms contain the product $\frac{\delta F}{\delta \mathbf{u}}\cdot\frac{\delta H}{\delta \mathbf{u}}^{\perp}$ and are hence by themselves antisymmetric. In particular, this also implies that to recover an advecting velocity in the velocity space $\mathbb{V}_u$, we have to apply $\mathbb{U}$ twice in each integral: once for $\frac{\delta H}{\delta \mathbf{u}}$ to recover the advecting velocity $\mathbf{u}$, and once for $\frac{\delta F}{\delta \mathbf{u}}$ to maintain antisymmetry. Note that as $\mathbb{U}$ corresponds to division by D, we introduce, for the purpose of consistency, an additional depth term D wherever $\mathbb{U}$ is not applied to a velocity space element of flux form $P_{\mathbb{V}_u}(D\mathbf{u})$. Thus, the curl part of the discrete Lie derivative (2.24) is given in the Hamiltonian variational setting by

$$-\left\langle \nabla^{\perp}\left(D\mathbb{U}\left(D,\tfrac{\delta F}{\delta \mathbf{u}}\right)\cdot\mathbb{U}\left(D,\tfrac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right), \mathbf{u}\right\rangle + \int_{\Gamma}\left[\!\left[D\mathbb{U}\left(D,\tfrac{\delta F}{\delta \mathbf{u}}\right)\cdot\mathbb{U}\left(D,\tfrac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right]\!\right]\mathbf{n}^{\perp}\cdot\tilde{\mathbf{u}}\, dS \tag{2.28}$$

$$+\left\langle D\mathbb{U}\left(D,\frac{\delta F}{\delta \mathbf{u}}\right), f\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right\rangle. \tag{2.29}$$

Note that with this setup, unlike (2.27), the advecting velocity $\mathbb{U}(\mathbf{u}, D)$ is the same for depth and velocity advection. Altogether, the full almost Poisson bracket including upwinding for depth and velocity advection is given in the Definition below.

Definition 9. *The upwind stabilised almost Poisson bracket is given by*

$$\{F,H\} \coloneqq -\left\langle D\mathbb{U}\left(D,\frac{\delta F}{\delta \mathbf{u}}\right), f\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right\rangle \tag{2.30}$$

$$+\left\langle \nabla^{\perp}\left(D\mathbb{U}\left(D,\tfrac{\delta F}{\delta \mathbf{u}}\right)\cdot\mathbb{U}\left(D,\tfrac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right), \mathbf{u}\right\rangle - \int_{\Gamma}\left[\!\left[D\mathbb{U}\left(D,\tfrac{\delta F}{\delta \mathbf{u}}\right)\cdot\mathbb{U}\left(D,\tfrac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right]\!\right]\mathbf{n}^{\perp}\cdot\tilde{\mathbf{u}}\, dS \tag{2.31}$$

$$-\left\langle D\,\mathbb{U}\left(D,\frac{\delta F}{\delta \mathbf{u}}\right), \nabla\frac{\delta H}{\delta D}\right\rangle + \int_{\Gamma}\left[\!\left[\frac{\delta H}{\delta D}\mathbb{U}\left(D,\frac{\delta F}{\delta \mathbf{u}}\right)\right]\!\right]\tilde{D}\, dS \tag{2.32}$$

$$+\left\langle D\,\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right), \nabla\frac{\delta F}{\delta D}\right\rangle - \int_{\Gamma}\left[\!\left[\frac{\delta F}{\delta D}\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right)\right]\!\right]\tilde{D}\, dS. \tag{2.33}$$

Checking for antisymmetry, we find that the Coriolis term in line (2.30) as well as the

velocity advection terms in line (2.31) are antisymmetric by themselves, while the first and second terms respectively in (2.32) and (2.33) form antisymmetric pairs. Using this bracket, we finally arrive at a system of ODEs of the form

$$\begin{aligned}\langle \mathbf{w}, \mathbf{u}_t \rangle =& \left\langle \nabla^\perp \left(D\mathbb{U}\left(D, \mathbf{w}\right) \cdot \mathbb{U}\left(D, \mathbf{F}\right)^\perp \right), \mathbf{u} \right\rangle - \int_\Gamma \left[\!\left[D\mathbb{U}\left(D, \mathbf{w}\right) \cdot \mathbb{U}\left(D, \mathbf{F}\right)^\perp \right]\!\right] \mathbf{n}^\perp \cdot \tilde{\mathbf{u}} \, dS \\ & - \left\langle D\, \mathbb{U}\left(D, \mathbf{w}\right), \nabla \tfrac{\delta H}{\delta D} \right\rangle + \int_\Gamma \left[\!\left[\tfrac{\delta H}{\delta D} \mathbb{U}\left(D, \mathbf{w}\right) \right]\!\right] \tilde{D} \, dS - \left\langle D\mathbb{U}\left(D, \mathbf{w}\right), f\mathbb{U}\left(D, \mathbf{F}\right)^\perp \right\rangle, \quad (2.34)\end{aligned}$$

$$\langle \phi, D_t \rangle = + \left\langle D\, \mathbb{U}\left(D, \mathbf{F}\right), \nabla \phi \right\rangle - \int_\Gamma \left[\!\left[\phi \mathbb{U}\left(D, \mathbf{F}\right) \right]\!\right] \tilde{D} \, dS, \quad (2.35)$$

for test functions $\mathbf{w} \in \mathbb{V}_u$ and $\phi \in \mathbb{V}_D$.

In view of the time discretisations to follow and for ease of notation, we rewrite the above bracket as

$$\{F, H\} = L_{\mathbb{U}(D, \frac{\delta H}{\delta \mathbf{u}})}\left(\mathbf{u}; D\mathbb{U}(D, \tfrac{\delta F}{\delta \mathbf{u}})\right) + F_{(D, \mathbb{U}(D, \frac{\delta H}{\delta \mathbf{u}}), \frac{\delta H}{\delta D})}\left(\mathbb{U}(D, \tfrac{\delta F}{\delta \mathbf{u}})\right) + L^D_{\mathbb{U}(D, \frac{\delta H}{\delta \mathbf{u}})}\left(D; \tfrac{\delta F}{\delta D}\right), \quad (2.36)$$

with velocity advection operator L corresponding to (2.31), forcing operator F to (2.30) together with (2.32), and depth advection operator L^D to (2.33). The choice of notation for advection is based on the usual notation for the Lie derivative $\pounds_v$, for advecting velocity v. L then corresponds to the divergence-free part of velocity advection – in the Lie derivative sense, noting that in our case the divergence part is contained in $\frac{\delta H}{\delta D}$; for details on Lie derivatives see Appendix B. Finally, the choice of notation for forcing F is simply to resemble L, in that the lowered terms indicate fields that act on $\mathbf{u}$, similarly to the advecting velocity in L. Note that since F does not explicitly contain the velocity field it is acting on, its only explicit argument in this notation is the test function $\mathbb{U}(D, \frac{\delta F}{\delta \mathbf{u}})$.

Remark 10. *Comparing the velocity stabilisation in the almost Poisson bracket based form* (2.34) *introduced above with the curl part of the discretised Lie derivative* (2.24)*, we find*

$$L_{\mathbf{u}}\big(\mathbf{u}; \mathbf{w}\big) \quad \longrightarrow \quad L_{\mathbb{U}(D, \mathbf{F})}\big(\mathbf{u}; D\mathbb{U}(D, \mathbf{w})\big), \quad (2.37)$$

that is we replaced the advecting velocity $\mathbf{u}$ *by the flux-recovered velocity* $\mathbb{U}(D, \mathbf{F})$*, while the advected velocity remains unchanged. Further, the test function* $\mathbf{w}$ *is replaced by a discrete multiplication and division of* $\mathbf{w}$ *by* D*. Similarly, for advection in* D *the advecting velocity is now also given by* $\mathbb{U}(D, \mathbf{F})$ *compared to standard DG upwinding* $L^D_{\mathbf{u}}(D; \phi)$ *given by* (1.128)*:*

$$L^D_{\mathbf{u}}(D; \phi) \quad \longrightarrow \quad L^D_{\mathbb{U}(D, \mathbf{F})}(D; \phi). \quad (2.38)$$

For more details on the significance of $\mathbb{U}(D, \mathbf{F})$*, see Appendix B. The latter describes a bracket based on momentum* μ *instead of the velocity* $\mathbf{u}$*, and we find that in this case the variations in* μ *for the Hamiltonian* H *and functional* $F = \langle \mathbf{w}, \mathbf{u} \rangle$ *are given by*

$$\frac{\delta H}{\delta \mu} = \mathbb{U}\left(D, \frac{\delta H}{\delta \mathbf{u}}\right), \qquad \frac{\delta F}{\delta \mu} = \mathbb{U}\left(D, \frac{\delta F}{\delta \mathbf{u}}\right), \tag{2.39}$$

noting that in the momentum based formulation, the velocity $\mathbf{u}(\mu, D)$ *is given as a function of momentum and depth.*

Finally, for the average depth upwind formulation (2.20)*, we can in turn view* $\mathbf{F}/D$ *as the advecting velocity (noting that* $\langle \nabla\phi, \mathbf{F} \rangle = \langle \nabla\phi, (\mathbf{F}/D)D \rangle$*). The same holds true for the velocity advection form* (2.27)*, where we find*

$$L_{\mathbf{u}}(\mathbf{u}; \mathbf{w}) \quad \longrightarrow \quad L_{\mathbf{F}/D}(\mathbf{u}; \mathbf{w}), \tag{2.40}$$

i.e. the advected velocity and test function remain the same, while the advecting velocity is replaced by $\mathbf{F}/D$*.*

While the recovered velocity advection method (2.37) *is newly presented here, the flux based velocity advection method* (2.40) *was considered previously in [NSC16]. The recovered depth advection method* (2.38) *or similar versions of it (such as the flux averaged form* (2.20)*) has not been considered in the finite element literature before, and in the numerical results section we will focus on this method rather than the velocity stabilisation aspect presented in this section.*

2.2 Numerical results

In the previous section we introduced the upwind stabilised bracket given in Definition 9, which is based on the simplified standard bracket (2.1) and extends it to include upwinding in the depth and velocity fields. To demonstrate conservation of energy, we additionally use an energy conserving time discretisation, thus expecting energy conservation up to machine precision*. Before moving on to the test cases, we describe the solver method for the resulting nonlinear system of equations, and review similar space discretisations with which the above discretisation will be compared.

* As noted in Section 1.2.4, we strictly speaking expect energy conservation only up to the nonlinear solver strategy and solver tolerance. In this and the following two chapters, we nonetheless find that energy conservation up to machine precision is often indeed achieved.

2.2.1 Nonlinear system

The space discretised equations (2.34) - (2.35) arising from the upwind stabilised almost Poisson bracket given in Definition 9 can be time discretised as outlined in Section 1.2.4. The resulting Hamiltonian variations are given by

$$\overline{\frac{\delta H}{\delta \mathbf{u}}} = \frac{1}{3} P_{\mathbb{V}_u}\big(D^n\mathbf{u}^n + \frac{1}{2}D^n\mathbf{u}^{n+1} + \frac{1}{2}D^{n+1}\mathbf{u}^n + D^{n+1}\mathbf{u}^{n+1}\big), \tag{2.41}$$

$$\overline{\frac{\delta H}{\delta D}} = P_{\mathbb{V}_D}\big(\frac{1}{6}(|\mathbf{u}^n|^2 + \mathbf{u}^n\cdot\mathbf{u}^{n+1} + |\mathbf{u}^{n+1}|^2) + g(\frac{1}{2}(D^n + D^{n+1}) + b)\big). \tag{2.42}$$

Remark 11. *Since* $\overline{\frac{\delta H}{\delta \mathbf{u}}}$ *is not of a simple flux form* $D\mathbf{u}$ *anymore, we find that the point-wise relation* (2.6) *for our velocity recovering operator does not hold anymore for this time discretisation, and we have to revert to the defining relation of* $\mathbb{U}$*, now given by*

$$\left\langle \frac{1}{2}(D^n + D^{n+1})\mathbf{v}, \mathbb{U} \right\rangle = \langle \mathbf{v}, \mathbf{F}\rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u. \tag{2.43}$$

Note that in accordance with (1.116)*, we choose a midpoint time average for* D *here since in view of the Poisson system, this relation is part of the transformation* J*. For* $\overline{\frac{\delta H}{\delta \mathbf{u}}}$*, the above relation* (2.43) *is hence given by*

$$\langle \tfrac{1}{2}(D^n + D^{n+1})\mathbf{v}, \mathbb{U}\rangle = \tfrac{1}{3}\langle \mathbf{v}, D^n\mathbf{u}^n + \tfrac{1}{2}D^n\mathbf{u}^{n+1} + \tfrac{1}{2}D^{n+1}\mathbf{u}^n + D^{n+1}\mathbf{u}^{n+1}\rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.44}$$

replacing the point-wise version (2.6)*.*

Writing $\bar{D} = \frac{1}{2}(D^n + D^{n+1})$, $\bar{\mathbf{u}} = \frac{1}{2}(\mathbf{u}^n + \mathbf{u}^{n+1})$, and $\bar{\mathbb{U}}$ for the solution to (2.44), we arrive at a fully discretised set of nonlinear equations of the form

$$\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n\rangle = \Delta t\Big(L_{\bar{\mathbb{U}}}\big(\bar{\mathbf{u}}; \bar{D}\mathbb{U}(\bar{D}, \mathbf{w})\big) + F_{(\bar{D}, \bar{\mathbb{U}}, \overline{\frac{\delta H}{\delta D}})}\big(\mathbb{U}(\bar{D}, \mathbf{w})\big)\Big) \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{2.45}$$

$$\langle \phi, D^{n+1} - D^n\rangle = \Delta t L^D_{\bar{\mathbb{U}}}(\bar{D}; \phi) \qquad \forall \phi \in \mathbb{V}_D, \tag{2.46}$$

to be solved for $\mathbf{u}^{n+1}$, D^{n+1}. Note that the Poisson integrator also requires upwinding using $\bar{\mathbf{u}}$, as the upwinded fields $\tilde{D}$ and $\tilde{u}$ are part of the transformation J (cf (1.129)).

To solve for the nonlinear system of equations (2.45) - (2.46), we start from the governing equations in a residual formulation:

$$0 = \mathbf{R}(\mathbf{z}^{n+1}; (\mathbf{w}, \phi)^T) = (R_{\mathbf{u}}, R_D)^T, \tag{2.47}$$

for $R_{\mathbf{u}}$, R_D defined as the difference of the left-hand and right-hand sides of (2.45) and (2.46), respectively. We then aim to solve the above using a quasi-Newton type iterative method, where we obtain the next time value $\mathbf{z}^{n+1,k+1}$ given the old value $\mathbf{z}^n$ and the latest guess for the next time value $\mathbf{z}^{n+1,k}$. Using an increment $\delta\mathbf{z} \coloneqq \mathbf{z}^{n+1,k+1} - \mathbf{z}^{n+1,k}$, we find

$$\begin{aligned} 0 = \mathbf{R}(\mathbf{z}^{n+1}; (\mathbf{w}, \phi)^T) &\approx \mathbf{R}(\mathbf{z}^{n+1,k+1}; (\mathbf{w}, \phi)^T) \\ &= \mathbf{R}(\mathbf{z}^{n+1,k} + \delta\mathbf{z}; (\mathbf{w}, \phi)^T) \\ &= \mathbf{R}(\mathbf{z}^{n+1,k}; (\mathbf{w}, \phi)^T) + \langle \frac{\delta \mathbf{R}}{\delta \mathbf{z}} \, \delta\mathbf{z}, (\mathbf{w}, \phi)^T \rangle + O(\|\delta\mathbf{z}\|^2) \\ &\approx \mathbf{R}(\mathbf{z}^{n+1,k}; (\mathbf{w}, \phi)^T) + \langle \frac{\delta \mathbf{R}}{\delta \mathbf{z}} \, \delta\mathbf{z}, (\mathbf{w}, \phi)^T \rangle, \end{aligned} \tag{2.48}$$

and hence

$$-\mathbf{R}(\mathbf{z}^{n+1,k}; (\mathbf{w}, \phi)^T) \approx \langle \frac{\delta \mathbf{R}}{\delta \mathbf{z}} \, \delta\mathbf{z}, (\mathbf{w}, \phi)^T \rangle. \tag{2.49}$$

To treat the right-hand side variational derivative, we first simplify by considering the residual $\mathbf{R}'$ derived from a weak form of the continuous rotating shallow water equations (1.71) - (1.72) instead of $\mathbf{R}$, thus avoiding projections introduced in the energy-conserving framework. Further, we revert to a Picard iteration method by linearising over a background state given by $(\mathbf{u}, D) = (\mathbf{0}, h)$, for constant reference height h. This allows us to apply a hybridised solver strategy for the resulting system of equations to be solved in each iteration; for more details on this strategy see [GMHC20] and e.g. [SGC18]. The linearised residual $\mathbf{R}'(\mathbf{z}^{n+1})$ is given by

$$\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n \rangle + \langle f\mathbf{w}, \frac{\Delta t}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n)^\perp \rangle - \langle \nabla \cdot \mathbf{w}, g\Delta t\big(\frac{1}{2}(D^{n+1} + D^n) + b\big) \rangle, \tag{2.50}$$

$$\langle \phi, D^{n+1} - D^n \rangle + \langle \phi, \nabla \cdot (h\frac{\Delta t}{2}(\mathbf{u}^{n+1} + \mathbf{u}^n)) \rangle, \tag{2.51}$$

and applying its variational derivative to $\delta\mathbf{z}$, we arrive at a right-hand side of the form

$$\langle \frac{\delta \mathbf{R}}{\delta \mathbf{z}} \, \delta\mathbf{z}, (\mathbf{w}, \phi)^T \rangle \approx \begin{pmatrix} \langle \delta\mathbf{u}, \mathbf{w} \rangle + \frac{\Delta t}{2} \langle f\delta\mathbf{u}^\perp, \mathbf{w} \rangle - \frac{\Delta t}{2} \langle g\delta D, \nabla \cdot \mathbf{w} \rangle \\ \langle \delta D, \phi \rangle + \frac{\Delta t}{2} \langle h \nabla \cdot \delta\mathbf{u}, \phi \rangle \end{pmatrix}. \tag{2.52}$$

Remark 12. *In order to consider the left-hand side of* (2.49)*, we need to find a way to treat the velocity recovery operator applied to test functions, i.e.* $\mathbb{U}(\bar{D}, \mathbf{w})$*, in the fully discretised momentum equation* (2.45)*. Noting that* $\mathbb{U}$ *corresponds to a discrete division*

by $\bar{D}$, this can be done by using test functions weighted by $\bar{D}$. That is, to solve for general $G\big(\mathbb{U}(\bar{D},\mathbf{w})\big)$, we first find $\mathbf{u}$ such that

$$\langle \mathbf{u}, \bar{D}\mathbf{v}\rangle = G(\mathbf{v}) \qquad\qquad \forall \mathbf{v} \in \mathbb{V}_u. \tag{2.53}$$

Then in particular, for any given test function $\mathbf{w}$, we have

$$G\big(\mathbb{U}(\bar{D},\mathbf{w})\big) = \langle \mathbf{u}, \bar{D}\mathbb{U}(\bar{D},\mathbf{w})\rangle = \langle \mathbf{u}, \mathbf{w}\rangle, \tag{2.54}$$

where we used (2.53) *for the first equality and the time-discrete defining relation* (2.43) *of $\mathbb{U}$ for the second one, noting that here $\mathbf{w}$ plays the role of $\mathbf{F}$ in the defining relation, while $\mathbf{u}$ corresponds to a particular choice of test function $\mathbf{v}$.*

Finally, the residual $-\mathbf{R}(\mathbf{z}^{n+1,k};(\mathbf{w},\phi)^T)$ can be calculated directly using the discretised equations (2.45) and (2.46). For the momentum equation, we first find forcing and advection velocities $\mathbf{u}^f, \mathbf{u}^a$ given by

$$\langle \mathbf{u}^f, \bar{D}\mathbf{v}\rangle = \Delta t F_{(\bar{D},\bar{\mathbb{U}},\frac{\delta H}{\delta D})}(\mathbf{v}) \qquad\qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.55}$$

$$\langle \mathbf{u}^a, \bar{D}\mathbf{v}\rangle = \Delta t L_{\bar{\mathbb{U}}}\big(\bar{\mathbf{u}}; \bar{D}\mathbf{v}\big) + \langle \mathbf{u}^n, \bar{D}\mathbf{v}\rangle \qquad\qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.56}$$

that is $\mathbf{u}^f$ corresponds to the additional velocity induced by forcing, while $\mathbf{u}^a$ corresponds to $\mathbf{u}^n$ after advection. Next, given that (2.55) and (2.56) hold, we find that in particular they hold for $\mathbf{v} = \mathbb{U}(\bar{D},\mathbf{w}) \in \mathbb{V}_u$ (as described in Remark 12), so that for any $\mathbf{w} \in \mathbb{V}_u$, (2.45) can be reformulated to

$$R_{\mathbf{u}}(\mathbf{z}^{n+1,k};\mathbf{w}) = \langle \mathbf{w}, \mathbf{u}^{n+1,k} - \mathbf{u}^n\rangle - \Delta t\Big(L_{\bar{\mathbb{U}}}\big(\bar{\mathbf{u}}; \bar{D}\mathbb{U}(\bar{D},\mathbf{w})\big) + F_{(\bar{D},\bar{\mathbb{U}},\frac{\delta H}{\delta D})}\big(\mathbb{U}(\bar{D},\mathbf{w})\big)\Big) \tag{2.57}$$

$$= \langle \mathbf{w}, \mathbf{u}^{n+1,k}\rangle - \langle \bar{D}\mathbb{U}(\bar{D},\mathbf{w}), \mathbf{u}^n\rangle$$

$$- \Delta t L_{\bar{\mathbb{U}}}\big(\bar{\mathbf{u}}; \bar{D}\mathbb{U}(\bar{D},\mathbf{w})\big) - \Delta t F_{(\bar{D},\bar{\mathbb{U}},\frac{\delta H}{\delta D})}\big(\mathbb{U}(\bar{D},\mathbf{w})\big) \tag{2.58}$$

$$= \langle \mathbf{w}, \mathbf{u}^{n+1,k}\rangle - \langle \bar{D}\mathbb{U}(\bar{D},\mathbf{w}), \mathbf{u}^a\rangle - \langle \bar{D}\mathbb{U}(\bar{D},\mathbf{w}), \mathbf{u}^f\rangle \tag{2.59}$$

$$= \langle \mathbf{w}, \mathbf{u}^{n+1,k} - \mathbf{u}^a - \mathbf{u}^f\rangle, \tag{2.60}$$

noting that we used the time-discrete defining relation (2.43) of $\mathbb{U}$ for the flux mass terms

(i.e. $\langle \bar{D}\mathbb{U}(\bar{D}, \mathbf{w}), \cdot \rangle$) of $\mathbf{u}^n$, $\mathbf{u}^a$ and $\mathbf{u}^f$. Similarly, for the residual in D we find

$$R_D(\mathbf{z}^{n+1,k}; \phi) = \langle \phi, D^{n+1,k} - D^a \rangle, \tag{2.61}$$

where D^a corresponds to D^n after advection and is solved for analogously to $\mathbf{u}^a$ in (2.56). Note that $\mathbf{u}^a + \mathbf{u}^f$ and D^a can be seen as a guess for the next iteration value $k+1$. Further, note that the residual $\mathbf{R}$ on the left-hand side of (2.49) is explicit in that it depends on the known values $\mathbf{z}^n$ and $\mathbf{z}^{n+1,k}$ only. To increase the method's robustness, we can instead also solve for a more implicit residual system of the form

$$\langle \mathbf{u}^a, \bar{D}\mathbf{v} \rangle = \Delta t L_{\bar{\mathbb{U}}}\big(\hat{\mathbf{u}}; \bar{D}\mathbf{v}\big) + \langle \mathbf{u}^n, \bar{D}\mathbf{v} \rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.62}$$

$$\langle D^a, \phi \rangle = \Delta t L^D_{\bar{\mathbb{U}}}\big(\hat{D}, \phi\big) + \langle D^n, \phi \rangle \qquad \forall \phi \in \mathbb{V}_D, \tag{2.63}$$

where we replaced the known advected time-averages $\bar{\mathbf{u}}, \bar{D}$ by implicit averages

$$\hat{\mathbf{u}} \coloneqq (\mathbf{u}^n + \mathbf{u}^a)/2, \quad \hat{D} \coloneqq (D^n + D^a)/2. \tag{2.64}$$

Further, D^a now also appears in the forcing term via a modified variation $\overline{\frac{\delta H}{\delta D}}$, given by

$$\overline{\frac{\delta H}{\delta D}} = P_{\mathbb{V}_D}\big(\frac{1}{6}(|\mathbf{u}^n|^2 + \mathbf{u}^n \cdot \mathbf{u}^{n+1,k} + |\mathbf{u}^{n+1,k}|^2) + g(\frac{1}{2}(D^n + D^a) + b)\big), \tag{2.65}$$

i.e. we replaced the known $D^{n+1,k}$ by D^a. Note that this implicit setup constitutes a different Picard iteration method, which, however, can be shown to converge to the same solution as the more explicit version (2.55) - (2.56). For the test cases used below, we will use this form for the higher resolution Galewsky test case. Further, note that the implicit setup cannot be implemented as readily for the discretisations without upwinding and the average upwinding form (2.20), respectively, since in this case the advected depth D is given implicitly in $\mathbf{F}$. We then have to solve for a mixed system in $\mathbf{F}^a$ and D^a, given by

$$\langle D^a, \phi \rangle = \Delta t \langle \phi, \nabla \cdot \mathbf{F}^a \rangle + \langle D^n, \phi \rangle \qquad \forall \phi \in \mathbb{V}_D, \tag{2.66}$$

$$\langle \mathbf{v}, \mathbf{F}^a - \frac{1}{3}\big(D^n\mathbf{u}^n + \frac{1}{2}D^n\mathbf{u}^{n+1} + \frac{1}{2}D^a\mathbf{u}^n + D^a\mathbf{u}^{n+1}\big) \rangle = 0 \qquad \forall \mathbf{v} \ \in \mathbb{V}_u, \tag{2.67}$$

for the non-upwinded case, and similarly for the averaged form (2.20).

This completes the fully energy-conserving scheme, with numerical test results presented in Section 2.2.3. The meshes, finite element discretisations, and solvers were implemented

using the automated finite element toolkit Firedrake[†] [RHM+16]. For all test cases in this chapter, the next to lowest (polynomial) order triangular finite element space triple (P_3, BDM_2, P_1^{DG}) was used. For test cases including quadrilateral cells, we refer to Chapters 3 and 4 below. The resulting systems of equations are solved using PETSc [BAA+19, BGMS97], noting that as mentioned above, a hybridised solver is used to solve for the mixed system in $(\delta\mathbf{u},\ \delta D)$. The fields depicted in the plots below are projections from the underlying compatible spaces into first (polynomial) order DG and CG spaces.

Remark 13. *While this section focuses on space discretisations, it is worth noting that time discretisations other than the Poisson integrator can be used. Considering implicit discretisations (given the context of numerical weather prediction), we find that a simpler time discretisation, albeit non-energy conserving, would be to use a midpoint rule for both* $\mathbf{u}$ *and* D*. The point-wise relation* (2.6) *then holds again if we further time-discretise the Hamiltonian variation in* $\mathbf{u}$ *as* $\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_u}(\bar{D}\bar{\mathbf{u}})$ *and the velocity recovery operator as before (*(2.43)*). The resulting left-hand side equations to be solved then read*

$$\langle \mathbf{u}^f, \bar{D}\mathbf{v}\rangle = \Delta t F_{(\bar{D},\bar{\mathbf{u}},\overline{\frac{\delta H}{\delta D}})}(\mathbf{v}) \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.68}$$

$$\langle \mathbf{u}^a, \bar{D}\mathbf{v}\rangle = \Delta t L_{\bar{\mathbf{u}}}\big(\bar{\mathbf{u}}; \bar{D}\mathbf{v}\big) + \langle u^n, \bar{D}\mathbf{v}\rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{2.69}$$

$$\langle D^a, \phi\rangle = \Delta t L^D_{\bar{\mathbf{u}}}(\bar{D}, \phi) + \langle D^n, \phi\rangle \qquad \forall \phi \in \mathbb{V}_D. \tag{2.70}$$

In particular, we find that in this case only one projection, i.e. $\overline{\frac{\delta H}{\delta D}}$*, occurs. Since it is a projection into the discontinuous Galerkin space* $\mathbb{V}_D$*, the additional cost of calculating it in each Picard iteration is low. In contrast, for no upwinding in* D *(e.g. as described in* (2.71) *below), this time discretisation leads to an additional Hamiltonian variation in* $\mathbf{u}$ *rather than* D *(appearing in the depth advection term), leading to a higher increase in computational cost as the underlying velocity space* $\mathbb{V}_u$ *is not a DG space (i.e. contains nodes on cell boundaries, leading to a non-block diagonal matrix). Further, for the averaged depth upwinding formulation* (2.20)*, we find that both the projection corresponding to the Hamiltonian variation in* $\mathbf{u}$ *and the projection* $\overline{\frac{\delta H}{\delta D}}$ *are required.*

2.2.2 Comparison to other discretisations

To test the newly introduced upwinding in D for energy conservation as well as the qualitative field development, we compare our upwinded formulation as given in Definition

[†] for further details, see [BMH+16, HMLH18, KM18, LHK17, MBM+16] or `http://firedrakeproject.org`

9 with one that includes upwinding in $\mathbf{u}$ only, and one that does not conserve energy. The former is given by an almost Poisson bracket with velocity terms equal to (2.31), but velocity forcing and depth advection terms of non-upwinded form, i.e.

$$\{F,H\} = L_{\mathbb{U}(D,\frac{\delta H}{\delta \mathbf{u}})}\left(\mathbf{u}; D\mathbb{U}(D,\frac{\delta F}{\delta \mathbf{u}})\right) + \left\langle \nabla\cdot\frac{\delta F}{\delta \mathbf{u}}, \frac{\delta H}{\delta D}\right\rangle - \left\langle \nabla\cdot\frac{\delta H}{\delta \mathbf{u}}, \frac{\delta F}{\delta D}\right\rangle \tag{2.71}$$

$$- \left\langle D\mathbb{U}\left(D,\frac{\delta F}{\delta \mathbf{u}}\right), f\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right\rangle. \tag{2.72}$$

Note that for this formulation, we will include the Coriolis term in the velocity advection step, since weighted test functions appear in both this term and the advection terms, while unweighted ones occur in the remaining forcing terms (consequently, the advection velocity $\mathbf{u}^a$ is calculated with weighted test functions, while the forcing velocity $\mathbf{u}^f$ uses unweighted ones).

As a non-energy conserving discretisation, we use standard DG upwinding for the depth field D, velocity upwinding of the form (2.24), and forcing equal to

$$\left\langle \nabla\cdot\mathbf{w}, \frac{1}{2}|\mathbf{u}|^2 + g(D+b)\right\rangle. \tag{2.73}$$

Note that in order to apply the same time discretisation to the non-energy conserving spatial discretisation, we need to rewrite the latter in terms of Hamiltonian variations and a bracket. It is given by

$$\begin{aligned}\{F,H\} =& L_{\mathbb{U}(D,\frac{\delta H}{\delta \mathbf{u}})}\left(\mathbf{u}; \frac{\delta F}{\delta \mathbf{u}}\right) + L^{D}_{\mathbb{U}(D,\frac{\delta H}{\delta \mathbf{u}})}\left(D; \frac{\delta F}{\delta D}\right) \\ & - \left\langle \frac{\delta F}{\delta \mathbf{u}}, f\mathbb{U}\left(D,\frac{\delta H}{\delta \mathbf{u}}\right)^{\perp}\right\rangle + \left\langle \nabla\cdot\frac{\delta F}{\delta \mathbf{u}}, \frac{\delta H}{\delta D}\right\rangle,\end{aligned} \tag{2.74}$$

and we find that as opposed to the energy-conserving upwinded version, the velocity advection operator as well as the Coriolis term are not antisymmetric in themselves (since $D\mathbb{U}$ is not applied to $\frac{\delta F}{\delta \mathbf{u}}$), and the depth advection and velocity forcing are not antisymmetric to each other.

2.2.3 Test cases

Having described the full discretisation as well as two other reference spatial discretisations, we proceed to our set of numerical tests. First, we consider two test cases to study

the qualitative field behaviour depending on whether or not upwinding is included. The first one is given by a wave in a periodic unit square mesh. Being more of an artificial test case, it serves as a proof of concept for introducing upwinding in the velocity field D, and thus in extension for the density field ρ in the context of the compressible Euler equations. The second one describes a breaking dam scenario, used to demonstrate the qualitative improvement due to upwinding in the presence of discontinuities in the depth field. For both of these test cases, we also consider the effect of upwinding quantitatively by using a "roughness" function of the form

$$DG(\mathbf{u}, D) \coloneqq \sqrt{\int_\Omega \left(\left| \frac{\partial \mathbf{u}}{\partial x} \right|^2 + \left| \frac{\partial \mathbf{u}}{\partial y} \right|^2 \right) dx + \frac{1}{\Delta x} \int_\Gamma [\![D]\!]^2 \, dS}, \tag{2.75}$$

noting that this choice reflects norms used in the stability analysis of DG methods (see e.g. [ABCM02]). After this, we consider more realistic standard spherical shallow water test cases for numerical weather prediction. Since these test cases only contain comparatively small gradients in the depth field, we will use them primarily to demonstrate that the new scheme produces the expected results in terms of field development as well as energy conservation.

Planar test cases

First, we consider a non-dimensionalised test case corresponding to a wave in a periodic unit square mesh as given in [MC14]. To test the qualitative field development, we consider the two energy-conserving versions with and without upwinding in D (and both including upwinding in $\mathbf{u}$) in a low resolution setup, and compare them to the non-energy conserving upwinded version in a high resolution setup. The initial conditions are given by

$$\begin{aligned} \mathbf{u}_0 &= (0,\ \sin(2\pi x)), \\ D_0 &= 1 + \frac{1}{4\pi} \frac{f}{g} \sin(4\pi y), \end{aligned} \tag{2.76}$$

with $(f,\ g) = (5,\ 5)$. The domain is divided into 32^2 and 128^2 squares for the low and high resolution setups, respectively, each of which in turn is divided into two triangles. The resulting fields for $\Delta t = 0.001$ and 1000 time steps (and $\Delta t = 0.0001$ and 10000 time steps, respectively for the high resolution setup), with 4 Picard iterations for each time step, are depicted in Figure 2.1.

We find that upwinding in D not only significantly reduces small scale perturbations in

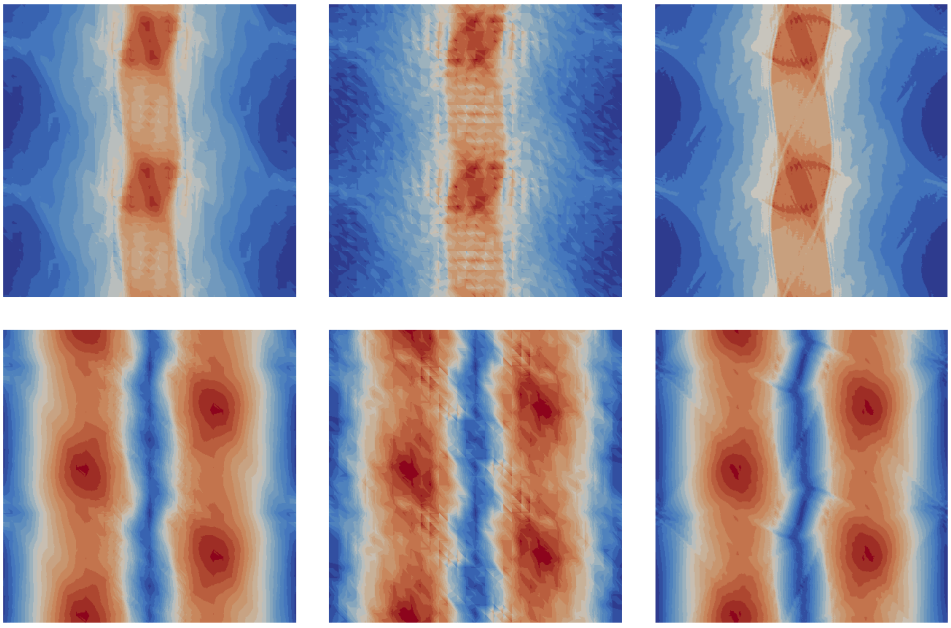


Figure 2.1: Field development for periodic unit square test case with low resolution energy conserving (EC) setups (with and without upwinding in D) and high resolution non-energy conserving (NEC) setup. All setups include upwinding in $\mathbf{u}$. Top row: depth fields for EC setup with and without upwinding, and NEC setup. Bottom row: velocity fields for EC setup with and without upwinding, and NEC setup. 2×32^2 cells, $\Delta t = 0.001$, 10^3 time steps, and 2×128^2 cells, $\Delta t = 0.0001$, 10^4 time steps for low and high resolution setups, respectively, with 4 Picard iterations for each time step. Depth field values 0.75 to 1.5 with contours every 0.05, velocity field magnitude values 0 to 1 with contours every 0.05.

the depth field, but also in the velocity field, with the upwinded version in good agreement with the high resolution setup in the resolved scales. Figure 2.4 depicts the development of $DG(\mathbf{u}, D)$ in time, confirming the reduction of small scale perturbations. As a side, if the non-energy conserving setup is run at low resolution, we further find that the resulting fields are in good agreement with the energy conserving upwinded scheme, except for smaller field magnitudes due to the loss of total energy.

To further test the effect of upwinding on the qualitative field development, we consider a breaking dam scenario as described in [Tor01]. The notable features in this test case consist of an outward moving shock wave, and an inward moving rarefaction wave which eventually collapses at the centre point, leading to another outward moving secondary shock wave. Note that the shallow water equations cover shock waves in their physically correct form only if the equations are considered in characteristic conservative form (i.e. evolving variables D and $\mathbf{u}D$). However, since we consider the shallow water equations in the context of numerical weather prediction, their primitive form (with variables D and $\mathbf{u}$) is used here. The primitive form still leads to waves that are a good approximation to the physically correct ones in case of a weak shock, that is for a depth ratio of size

approximately less than 3 (for more details, see [Tor01]). We therefore adjust the radial breaking dam test case given in [Tor01] to a weak shock by raising the ambient depth from 0.5 to 1.25 metres. The resulting initial conditions, for a square domain of side length $L = 40$ metres, are then given by

$$D(r,0) = \begin{cases} 2.5 \text{ m} & \text{if } r < 2.5, \\ 1.25 \text{ m} & \text{otherwise,} \end{cases} \tag{2.77}$$

$$|\mathbf{u}(r,0)| = 0 \text{ m/s},$$

for radius r with respect to the square's centre, and $(f,\ g) = (0\ \text{s}^{-1},\ 9.81\ \text{m/s}^2)$. The domain is divided into 200^2 cells, each of which in turn is divided into two triangles. To reflect the radially symmetric nature of this test case, the diagonals used to divide each cell are chosen such that the resulting mesh is symmetric with respect to the vertical and horizontal axes crossing the square's centre. Results for the depth field are depicted in Figures 2.2 and 2.3. The former shows 1D cross-sections along the line $x = 0$ at times 1.2 and 3 seconds for the energy conserving setups with and without upwinding in D, respectively (and both again including upwinding in $\mathbf{u}$). After 1.2 seconds, the inward travelling rarefaction wave has nearly reached the domain's centre, while at 3 seconds we can see the secondary shock wave following behind the initial one. Figure 2.3 depicts a top view of the depth field in the square's lower half after 4 seconds, with the left quadrant corresponding to the non-upwinded scheme, while the right quadrant does to the upwinded one. We find that since neither scheme is total variation diminishing, both of them produce oscillations near the shock waves (in particular the initial wave). However, these are smaller in the upwinded scheme (see e.g. top row, Figure 2.2), and we further find that small scale perturbations eventually corrupt most of the wave when the non-upwinded scheme is used (bottom row, Figure 2.2, and Figure 2.3). To remove the oscillations completely, we believe that a simple application of finite-volume/discontinuous Galerkin techniques is not likely to be possible in the compatible finite element framework. This is because our approach combines partially continuous and fully discontinuous finite element spaces, and characteristic variables (necessary for flux limiting) will become a combination of both. Instead, we believe that a flux-corrected transport method [KLT12] or residual distribution method [Abg06] would be necessary to remove oscillations. These additional fluxes would be likely to diffuse energy, but the Hamiltonian framework would still ensure that they were the only source of dissipation in the scheme. Aside from the compatible framework, an energy conserving DG method for the shallow water equations

can be found e.g. in [WWGK17]. The latter also includes a dam break scenario, with oscillations further mitigated by reverting to an entropy stable scheme, where the energy is dissipated across jumps.

As for the first square test case, the qualitative field analysis is confirmed by the time evolution of the "roughness" function (2.75), with larger values for the non-upwinded scheme compared to the upwinded one, as well as a steeper slope over the last 1000 time steps (i.e. in the two shock wave regime).

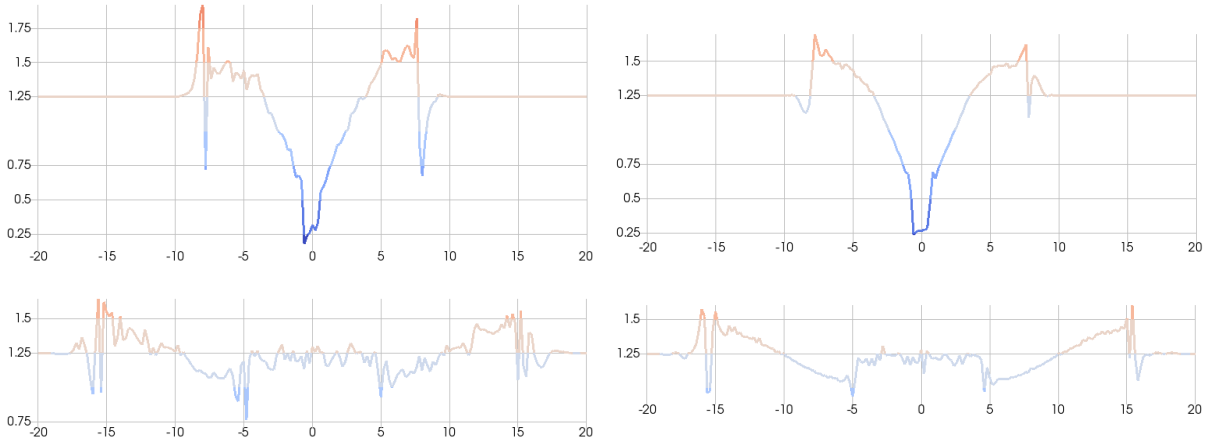


Figure 2.2: Depth field 1D cross-sections along line $x = 0$ for breaking dam test case, for energy conserving setup with (right column) and without (left column) upwinding in D. Top and bottom rows depict field after 1.2 and 3 seconds, respectively. Simulations were run with 2×200^2 cells, $\Delta t = 0.001$ s, 6 Picard iterations per time step.

Remark 14. *Next to the upwind scheme used so far in the numerical results section, which relies on the velocity recovery operator* $\mathbb{U}$*, we also introduced another possible depth upwinding method given by* (2.20)*, which is based on the average value avg*$(\mathbf{F}/D)$*. Together with the velocity advection operator* (2.40)*, we can use it to construct an antisymmetric bracket which relies on* $\mathbf{F}/D$ *as the advecting velocity. We find that while the energy conserving formulation based on this bracket also performs well in the unit square wave test case* (2.76)*, it fails to converge in the breaking dam test case* (2.77)*. The failure occurs as the rarefaction wave approaches the square's centre, leading to a very steep depth field gradient closing into the centre point, with jump values reaching 0.6 across facets. In this case, the approximation* $D^- \approx D^+$ *does not hold, likely causing the failure of convergence.*

Spherical test cases

Next, we consider more realistic spherical test cases. Since the depth field development in these cases is generally smoother than in the previous test cases, we anticipate the quali-

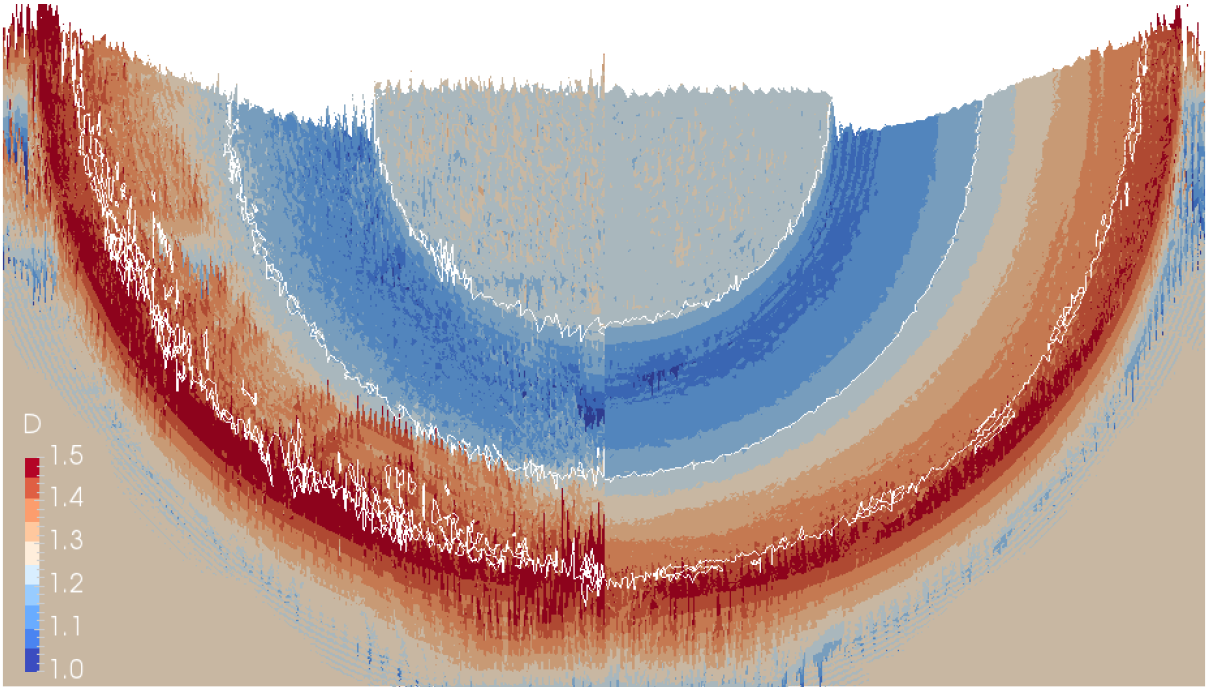


Figure 2.3: Top view of lower half of domain for breaking dam test case with shock wave after 4 seconds, for the energy conserving setup without (left quadrant) and with (right quadrant) upwinding. White contour lines depict the depth field projected into the continuous space CG_2, at depths 1.2 m, 1.3 m, and 1.4 m. 2×200^2 cells, $\Delta t = 0.001$ s, 6 Picard iterations per time step.

tative difference between the upwinded and non-upwinded versions to be small. Thus, the main purpose of these tests is to demonstrate an improved energy conservation as well as a qualitative field behaviour close to the projection-free non-energy conserving version.

To validate the new upwinded scheme's energy conservation as well as consistency, we use the second of the standard Williamson spherical test cases given in [WDH+92], which describes a steady state scenario. Additionally, to compare the schemes with respect to their energy conservation properties as well as field development, we use the fifth test of the aforementioned test series, corresponding to flow past a mountain, as well as the Galewsky barotropic instability test case as described in [GSP04]. In the Williamson 2 test case, the initial conditions are given by

$$\mathbf{u} = u_0(-y, x, 0)/a, \tag{2.78}$$

$$D = h - (a\Omega u_0 + u_0^2/2)\frac{z^2}{ga^2}, \tag{2.79}$$

for a sphere of radius $a = 6371220$ m, with rotation rate $\Omega = 7.292 \times 10^{-5}$ s^{-1} (noting that $f = 2\Omega z/a$), and gravitational acceleration $g = 9.810616$ m/s^2. The mean height

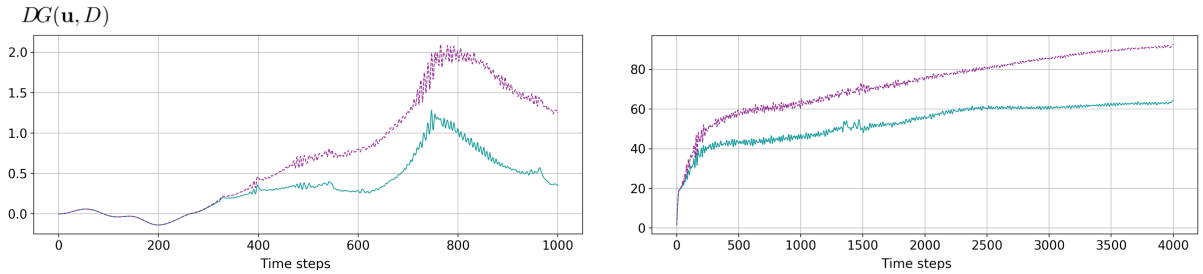


Figure 2.4: Development of roughness function DG (2.75) in time, for periodic unit square wave test case (left) and breaking dam test case (right), with energy conserving setups with (cyan) and without (dashed purple) upwinding in D.

and wind speed are given by $h = 3000$ m and $u_0 = \frac{2\pi a}{12}$ m/day. The simulation is run for 50 days, with a time step of $\Delta t = 50$ s, and 4 Picard iterations for each time step. The mesh is given by an icosahedral triangulation, where refinement level 0 corresponds to 20 triangles. For every higher level, each triangle is refined to 4 triangles (so that each increase corresponds to halving the cell side length Δx). The resulting relative energy error development, as well as the L^2 depth field error, averaged over the last 1000 time steps and for different refinement levels, are depicted in Figure 2.5.

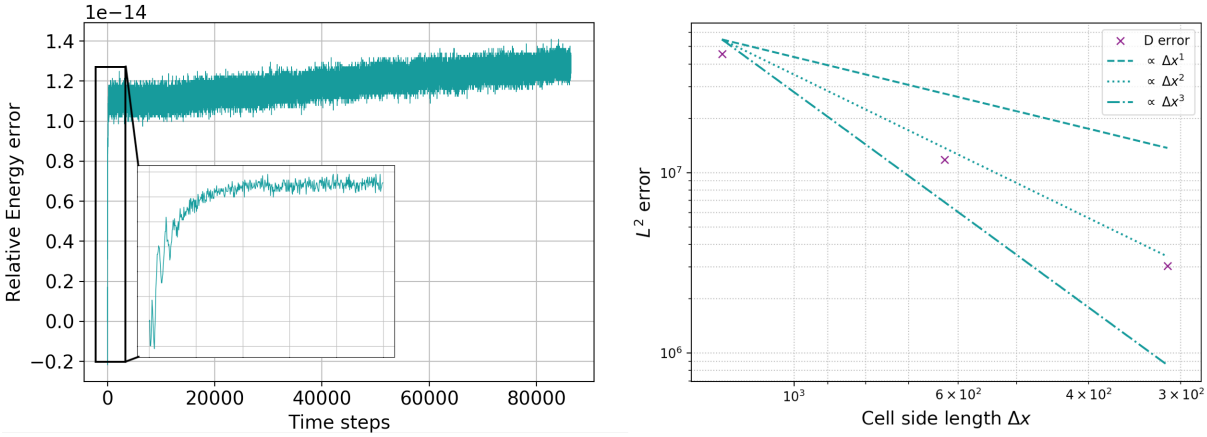


Figure 2.5: Left: relative energy error development for Williamson 2 test case, using fully energy conserving upwinded discretisation (2.45) - (2.46). Mesh refinement level 5, $\Delta t = 50$s, with 4 Picard iterations for each time step (window: first 500 time steps). Right: L^2 depth field error averaged over the last 1000 time steps for refinement levels 3 to 5.

As expected, we find that energy is conserved up to machine precision throughout the simulation, with the initial increase likely due to the simplified variational derivative of the residual $\mathbf{R}$. Further, the L^2 depth field error convergence as we refine the mesh also matches the expected second order rate (the same holding true for the velocity field).

Next, we consider the fifth Williamson test, corresponding to unsteady flow over a mountain. The initial conditions are given by

$$\mathbf{u} = u_0(-y, x, 0)/a, \tag{2.80}$$

$$D = h - (a\Omega u_0 + u_0^2/2)\frac{z^2}{ga^2} - b, \tag{2.81}$$

$$b = b_0(1 - r/R), \tag{2.82}$$

where b describes the mountain's surface, for $R = \pi/9$, mountain height $b_0 = 2000$ m and r such that $r = \min(R, \sqrt{(\lambda - \lambda_c)^2 + (\theta - \theta_c)^2})$. $\lambda \in [-\pi, \pi]$ and $\theta \in [-\pi/2, \pi/2]$ denote longitude and latitude respectively, and the mountain's centre is chosen as $\lambda_c = -\pi/2$ and $\theta_c = \pi/6$. The mean height and wind speed are given by $h = 5960$ m and $u_0 = 20$ m/s. The simulation is run for 25 days, with a time step of $\Delta t = 50$ s. Since this test involves an unsteady nonlinear field evolution, we increase the number of Picard iterations to 8 for each time step in order to keep the relative energy error contribution of the nonlinear solver small. The resulting relative energy errors and potential vorticity fields for the energy conserving setup with and without upwinding as well as the non-energy conserving setup are given in Figures 2.6 and 2.7.

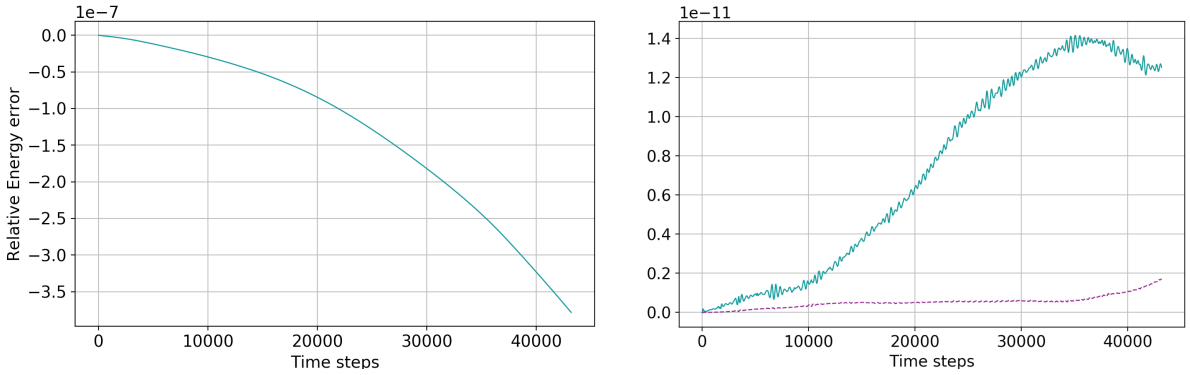


Figure 2.6: Relative energy error developments for Williamson 5 test case. Left: non-energy conserving setup. Right: energy conserving setup with (cyan) and without (dashed purple) upwinding in D. Mesh refinement level 5, $\Delta t = 50$ s, with 8 Picard iterations per time step.

Again, as expected the energy conserving formulations conserve energy up to a good degree, with a relative energy error of the order of 10^{-11}, four orders of magnitude smaller than the relative energy error for the non-energy conserving setup. Additionally, we find a practically identical field development for all three setups. This indicates that the new

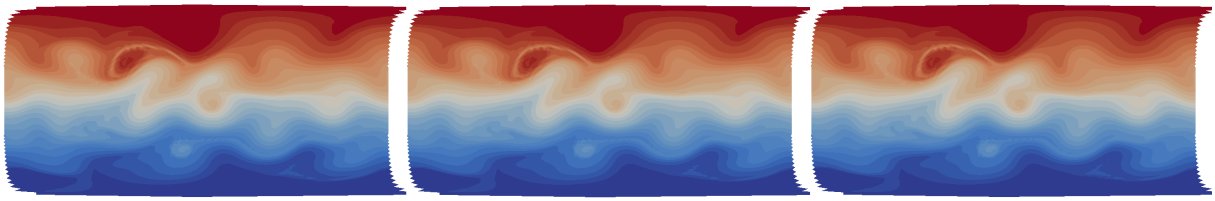


Figure 2.7: Potential vorticity fields after 25 days for Williamson 5 test case. Left to right: non-energy conserving setup, energy conserving setup with upwinding in D, energy conserving setup without upwinding in D. Mesh refinement level 5, $\Delta t = 50$ s, with 8 Picard iterations per time step. 30 contours, scale: -3×10^{-8} (blue) to 3×10^{-8} (red).

method for upwinding in D in a Poisson bracket framework also behaves as expected in a spherical domain, without adverse effects due to projections or the reformulated antisymmetric forcing term.

Remark 15. *For 4 Picard iterations, the relative energy error for the energy conserving setups increases to the order of* 10^{-9}*, while remaining at the order of* 10^{-7} *for the non-energy conserving space discretisation. In fact, at this stage the error originating from the nonlinear solver surpasses the decrease in error due to the energy-conserving time discretisation. We find that the midpoint rule time discretisation given in Remark 13, together with 4 Picard iterations, still yields relative energy errors of the order of* 10^{-9} *and* 10^{-7} *for the energy-conserving and non-energy conserving spatial discretisations, respectively. This demonstrates that the energy conserving spatial discretisations still lead to a significant improvement even for lower Picard iteration numbers and non-energy conserving time discretisations.*

Finally, we consider the Galewsky test case, simulating a barotropic instability. The initial conditions are given by a zonal flow u confined within latitudes $\theta_0 = \pi/7$ and $\theta_1 = 5\pi/14$, and a background depth field D in balance with u, perturbed by a localised bump D_p. The zonal flow is given by

$$u(\theta) = \frac{u_0}{e_n} e^{\frac{1}{(\theta-\theta_0)(\theta-\theta_1)}}, \tag{2.83}$$

for $u_0 = 80$ m/s, and normalising constant $e_n = \exp\big(-4/(\theta_1-\theta_0)^2\big)$. To reach a steady state depth field, we integrate the continuity equation (in spherical form), leading to a field of the form

$$gD(\theta) = gh_0 - \int_{-\frac{\pi}{2}}^{\theta} au(\theta')\big(f + \frac{\tan\theta'}{a} u(\theta')\big) d\theta', \tag{2.84}$$

where h_0 is chosen such that the global mean height is equal to 10km. Finally, the perturbation is given by

$$D_p = h_p \cos(\theta) e^{-(\lambda/\alpha)^2 - ((\theta_2-\theta)/\beta)^2}, \tag{2.85}$$

for $h_p = 120$ m , $\alpha = 1/3$, $\beta = 1/15$ and $\theta_2 = \pi/4$ (such that the perturbation is located directly in the zonal flow). The simulation is run for 6 days at mesh refinement level 6, with a time step of $\Delta t = 30$ s, and 8 Picard iterations for each time step. The resulting relative energy errors as well as vorticity fields for the three setups are depicted in Figures 2.8 and 2.9.

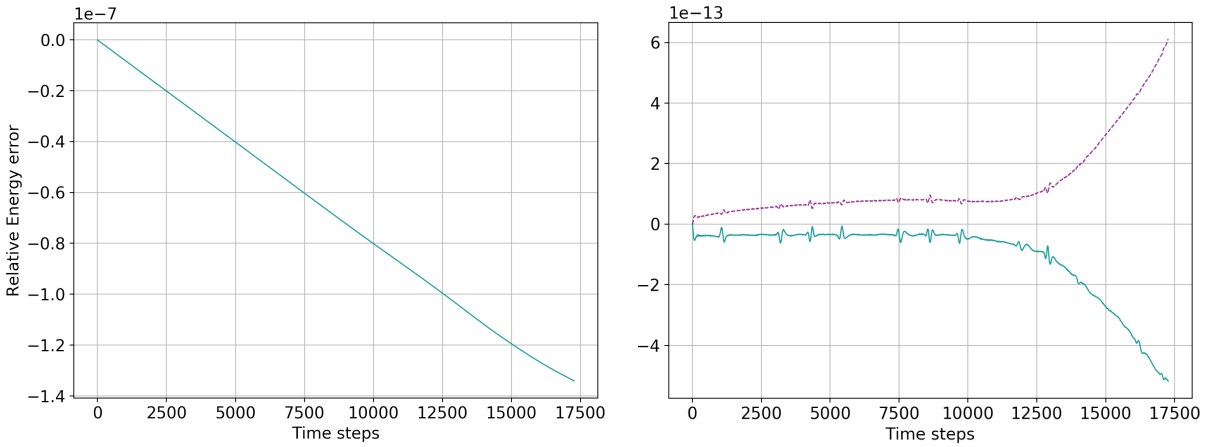


Figure 2.8: Relative energy error developments for Galewsky test case. Left: non-energy conserving setup. Right: energy conserving setup with (cyan) and without (dashed purple) upwinding in D. Mesh refinement level 6, $\Delta t = 30$ s, with 8 Picard iterations for each time step.

We find a similar behaviour to the Williamson 5 test case. With respect to the relative energy error, the two energy conserving setups outperform the non-energy conserving space discretisation by 6 orders of magnitude. Again, the field development is virtually the same for all setups.

Remark 16. *Next to energy, the non-upwinded space discretisation* (2.2) *also conserves enstrophy. To see this, we first derive a vorticity evolution equation using the momentum equation and the discretised vorticity relation* (1.112)*:*

$$\langle \eta, (qD)_t \rangle = -\langle \nabla^\perp \eta, \mathbf{u}_t \rangle, \tag{2.86}$$

$$\langle \nabla^\perp \eta, \mathbf{u}_t \rangle + \langle \nabla^\perp \eta, q\mathbf{F}^\perp \rangle - \langle \nabla \cdot \nabla^\perp \eta, \frac{\delta H}{\delta D} \rangle = \langle \nabla^\perp \eta, \mathbf{u}_t \rangle + \langle \nabla \eta, q\mathbf{F} \rangle = 0, \tag{2.87}$$

$$\Rightarrow \ \langle \eta, (qD)_t \rangle + \langle \eta, \nabla \cdot (q\mathbf{F}) \rangle = 0 \qquad \forall \eta \in W_0, \tag{2.88}$$

noting that $\nabla^\perp \eta \in \mathbb{V}_u$, $\nabla \cdot \nabla^\perp \eta = 0$, *and* q *and* η *are continuous. Conservation of enstrophy can then be shown using* $\eta = q$ *and* $D_t = -\nabla \cdot \mathbf{F}$ *(see Chapter 4). However, a*

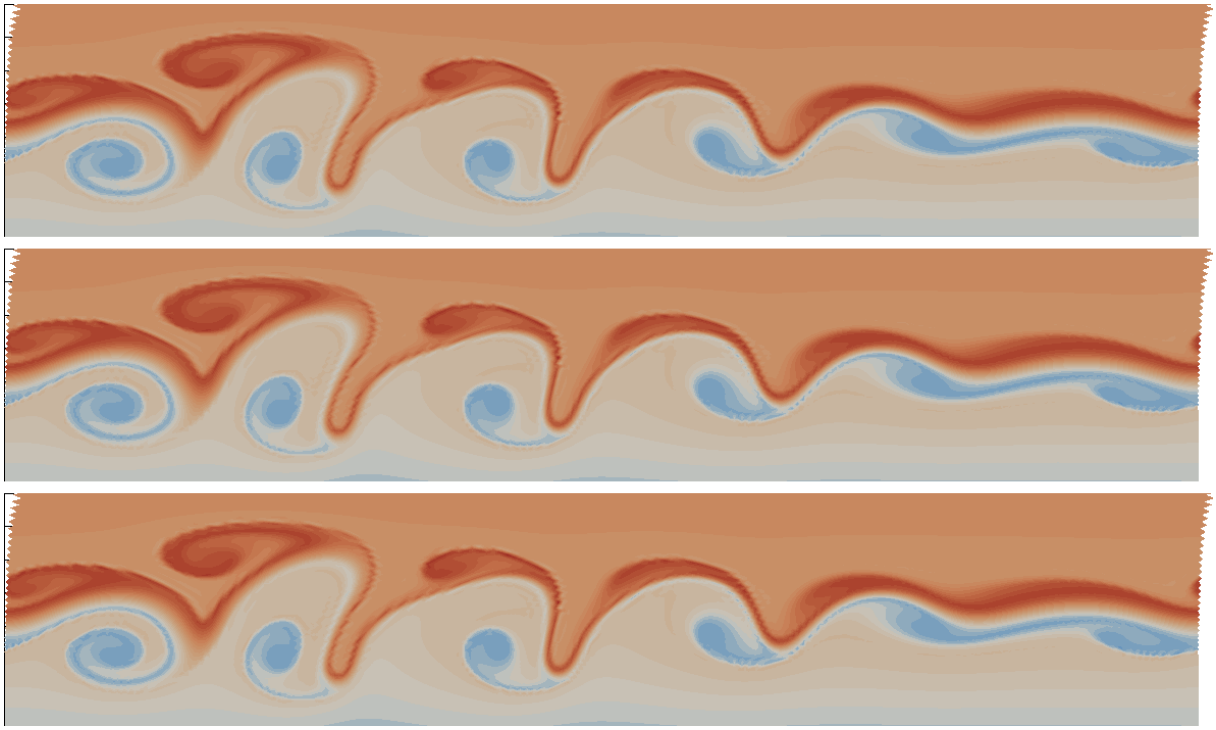


Figure 2.9: Potential vorticity fields between 10 and 80 degrees latitude after 6 days for the Galewsky test case. Top to bottom: non-energy conserving setup, energy conserving setup with upwinding in D, and without upwinding in D. Mesh refinement level 6, $\Delta t = 30$ s, with 8 Picard iterations for each time step. Contour lines every 1.25×10^{-9}, with negative values in blue, positive ones in red.

controlled dissipation of enstrophy may be desirable, since the enstrophy cascades to small scales, eventually accumulating at the grid scale. This effect was countered in [BC18] by including an SUPG method for the potential vorticity equation (2.88) *above, which implies enstrophy dissipation for a suitable SUPG parameter τ. In our case, upwinding in the velocity field $\mathbf{u}$ also dissipates enstrophy, noting that the vorticity advection term is derived from the velocity advection term. The changes due to our presented upwinding in the depth field may also alter the enstrophy evolution, since now the forcing term does not vanish in the vorticity equation, and further D_t is not equal to $-\nabla \cdot \mathbf{F}$ pointwise anymore. However, we find that upwinding in $\mathbf{u}$ with the underlying velocity space BDM_2 still controls the dissipation of enstrophy, with little difference in the dissipation rate whether or not upwinding in D is also included in the discretisation (see Figure 2.10). More details on the dissipation of enstrophy depending on the choice of upwinding for the velocity field can be found in [NC17a] and Chapter 4.*

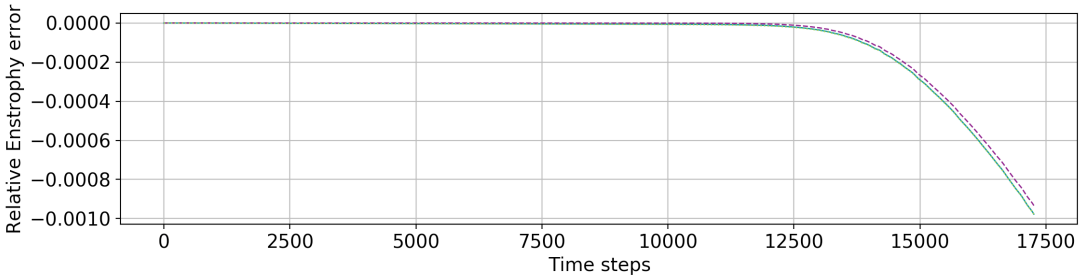


Figure 2.10: Relative enstrophy error developments for the Galewsky test case, for energy conserving setup with (cyan) and without (dashed purple) upwinding in D and non-energy conserving setup (dotted green).

2.3 Summary

In this chapter, we introduced an energy conserving space discretisation for the rotating shallow water equations that includes upwinding in the depth and velocity fields. The underlying almost Poisson bracket is based on one without upwinding, which is described in [MC14], and uses upwinding for the velocity field as formulated for the incompressible Euler equations in [NC17b]. Upwinding for the depth field in this context has been newly introduced here, and relies on the introduction of an additional operator to recover the velocity field from the Hamiltonian variation corresponding to the momentum flux. In our numerical tests, we confirmed the scheme's energy conservation property, with a relative energy error close to machine precision when coupled with an energy conserving time discretisation. In the spherical test cases, we showed that the fully upwinded energy conserving scheme behaves as expected for spherical domains, leading to a field development virtually identical to that of a corresponding non-energy conserving upwinded reference scheme, despite the additional projections that are required to achieve energy conservation. Further, as demonstrated in the unit square test cases, in the presence of large depth field gradients, the newly introduced upwinding in the depth field improves the field development compared to when upwinding is only applied for velocity, reducing small scale oscillations both in the depth and velocity fields.

The introduction of upwinding in the depth field was motivated by the development of a fully upwinded energy conserving space discretisation for the compressible Euler equations. While typical spherical shallow water scenarios feature a relatively small gradient in the depth field, the Euler equations often exhibit much stronger such gradients, therefore benefiting significantly from upwinding in the density field.

3

Thermal field stabilisation

In this chapter, we describe the incorporation of the SUPG advection operator as given in Definition 7 into the standard bracket (1.57), and present numerical results by means of the thermal rotating shallow water and 2D compressible Euler equations for a vertical slice of the atmosphere. For this purpose, we first recall that the bracket is given by

$$\begin{aligned}\{F,H\} \coloneqq &-\left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega}\times\frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla\cdot\frac{\delta F}{\delta \mathbf{u}}\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}}\right\rangle \\ &- \left\langle \frac{\delta F}{\delta \rho}, \nabla\cdot\frac{\delta H}{\delta \mathbf{u}}\right\rangle - \left\langle \frac{1}{\rho}\frac{\delta F}{\delta \theta}\nabla\theta, \frac{\delta H}{\delta \mathbf{u}}\right\rangle,\end{aligned} \tag{3.1}$$

which leads to an energy conserving space discretisation of the form (c.f. Section 1.2.4)

$$\langle \mathbf{w}, \mathbf{u}_t\rangle + \left\langle \mathbf{w}, \boldsymbol{\omega}\times\frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle - \left\langle \frac{\delta H}{\delta \rho}, \nabla\cdot\mathbf{w}\right\rangle - \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \mathbf{w}\right\rangle = 0 \qquad \forall \mathbf{w}\in\mathbb{V}_u, \tag{3.2}$$

$$\langle \phi, \rho_t\rangle + \left\langle \phi, \nabla\cdot\frac{\delta H}{\delta \mathbf{u}}\right\rangle = 0 \qquad \forall \phi\in\mathbb{V}_\rho, \tag{3.3}$$

$$\langle \gamma, \theta_t\rangle + \left\langle \gamma, \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\cdot\nabla\theta\right\rangle = 0 \qquad \forall \gamma\in\mathbb{V}_\theta, \tag{3.4}$$

together with a discretised diagnostic scalar vorticity equation, which in the 2D case is given by

$$\langle \eta, q\rho\rangle = -\left\langle \nabla^\perp\eta, \mathbf{u}\right\rangle + \left\langle\!\left\langle \eta, \mathbf{n}^\perp\cdot\mathbf{u}\right\rangle\!\right\rangle + \langle \eta, f\rangle \qquad \forall \eta\in\mathbb{V}_q. \tag{3.5}$$

$\langle\langle .,. \rangle\rangle$ denotes the L^2 inner product on $\partial\Omega$, and we recall that q is related to $\boldsymbol{\omega}$ via $\boldsymbol{\omega} = (q\rho)\mathbf{k}$, for the 2D domain's outward unit vector $\mathbf{k}$.

The incorporation of the SUPG stabilisation method follows that of the DG upwind stabilisation method in the previous chapter, in the sense that we first include the method in the standard bracket, then adjust the remaining bracket terms to ensure antisymmetry and therefore energy conservation. In Section 3.1, we present this mechanism for the thermal field transport equation. In Section 3.2, we review the treatment of the resulting nonlinear system of equations. While only mentioned briefly in the previous chapter, the latter section also contains a more thorough discussion on the choice of time discretisation. In particular, in view of the increased computational cost of the fully energy conserving scheme, an approximately energy conserving version based on a small departure from the Poisson time integrator is presented. In Section 3.3, we present numerical results based on energy conservation, qualitative field development, and the role of the number of Picard iterations when solving for the nonlinear system of equations. Finally, in Section 3.4, we summarise the chapter.

3.1 SUPG stabilisation in the discretised bracket

In this section, we incorporate the SUPG method, as described in Section 1.3.3, into the discretised standard bracket for the purpose of thermal field transport stabilisation. Before moving on to relevant definitions for the updated bracket, we outline the course of thought leading to it.

Without upwinding, the bracket is given by (3.1), such that $q \in \mathbb{V}_q$, $\rho \in \mathbb{V}_\rho$, $\theta \in \mathbb{V}_\theta$, and the Hamiltonian variations are functions of the corresponding finite element spaces. We aim to replace the corresponding thermal field advection equation (3.4) by the SUPG version (1.145), i.e.

$$\langle \gamma, \theta_t \rangle = -\left\langle \gamma, \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \cdot \nabla\theta \right\rangle \quad \rightarrow \quad \left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle = -L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}(\theta; \gamma_{\mathbf{u}}). \tag{3.6}$$

Note that since the advecting velocity in the non-upwinded thermal field advection equation is given by $\frac{\delta H}{\delta \mathbf{u}}/\rho$, we anticipate that this will also be the case when upwinding is applied. The non-upwinded version (to the left in (3.6)) was derived from choosing $F = \langle \gamma, \theta \rangle$ in the Poisson system (1.53), suggesting that for the upwinded version we

could instead pick a functional of the form

$$F = \left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta \right\rangle . \tag{3.7}$$

However, this is impractical, since then the time derivative of F will include a time derivative in the upwinding velocity $\mathbf{u}$. To avoid this, we move the corresponding upwinding part to the right-hand side:

$$\langle \gamma, \theta_t \rangle = -L^{\theta}_{\frac{\delta H}{\delta \rho}/\mathbf{u}}(\theta; \gamma_{\mathbf{u}}) - \left\langle \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle , \tag{3.8}$$

that is we will consider the upwind part $\tilde{\tilde{\gamma}}_{\mathbf{u}}$ of the time derivative term as part of the almost Poisson bracket. Including this term in the bracket, however, leads to further complications. First, a corresponding antisymmetric term needs to be added to the bracket, which will appear in the momentum equation's forcing part. Second, in view of the energy conserving Poisson time integrator (1.116), the bracket should be independent of time derivatives. This means that θ_t in the bracket should be replaced by another, time-derivative free term. Considering the upwinded equation (3.6), we find that θ_t depends on the advection operator L on the right-hand side as well as the upwinded test function contribution $\tilde{\tilde{\gamma}}_{\mathbf{u}}$ on the left-hand side, motivating Definitions 17 and 18 below.

Definition 17. *Given a velocity field $\mathbf{w} \in \mathbb{V}_u$, the thermal field advection operator $\theta^a(\mathbf{w})\colon \mathbb{V}_u \to \mathbb{V}_\theta$ is defined by*

$$\langle \gamma, \theta^a(\mathbf{w}) \rangle = -L^{\theta}_{\mathbf{w}/\rho}(\theta; \gamma_{\mathbf{u}}) \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{3.9}$$

Its adjoint operator $(\theta^a)^*(\gamma)\colon \mathbb{V}_\theta \to \mathbb{V}_u$ is given by

$$\langle (\theta^a)^*(\gamma), \mathbf{w} \rangle = -L^{\theta}_{\mathbf{w}/\rho}(\theta; \gamma_{\mathbf{u}}) \qquad \forall \mathbf{w} \in \mathbb{V}_u. \tag{3.10}$$

Definition 18. *Given a thermal field $\tau \in \mathbb{V}_\theta$, the SUPG recovery operator $\mathbb{S}_{\mathbf{u}}(\tau)\colon \mathbb{V}_\theta \to \mathbb{V}_\theta$ is defined by*

$$\langle \gamma, \mathbb{S}_{\mathbf{u}}(\tau) \rangle = \left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \tau \right\rangle \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{3.11}$$

Its adjoint operator $\mathbb{S}_{\mathbf{u}}^*(\gamma)\colon \mathbb{V}_\theta \to \mathbb{V}_\theta$ *is given by*

$$\mathbb{S}_{\mathbf{u}}^*(\gamma) = P_{\mathbb{V}_\theta}(\gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}), \tag{3.12}$$

for projection $P_{\mathbb{V}_\theta}$ *into the thermal field space* $\mathbb{V}_\theta$*.*

Note that both θ^a and $\mathbb{S}_{\mathbf{u}}$ implicitly also depend on the state $\mathbf{z} = (\mathbf{u}, \rho, \theta)$. Using these two operators, we can rewrite the upwinded thermal field advection equation (3.6) as

$$\langle \gamma, \mathbb{S}_{\mathbf{u}}(\theta_t)\rangle = \left\langle \gamma, \theta^a\Big(\frac{\delta H}{\delta \mathbf{u}}\Big)\right\rangle \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{3.13}$$

for any test function γ. Since both θ^a and $\mathbb{S}_{\mathbf{u}}$ map into $\mathbb{V}_\theta$, this implies $\theta^a = \mathbb{S}_{\mathbf{u}}(\theta_t)$ point-wise, and therefore $\theta_t = \mathbb{S}_{\mathbf{u}}^{-1}(\theta^a)$, provided $\mathbb{S}_{\mathbf{u}}^{-1}$ exists. This motivates a modified, upwinded θ variation part of the bracket according to

$$+\left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}}\right\rangle \qquad +L^\theta_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\Big(\theta; \Big(\frac{\delta H}{\delta \theta}\Big)_{\mathbf{u}}\Big) + \left\langle \Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\Big(\frac{\delta F}{\delta \mathbf{u}}\Big)\Big)\right\rangle \tag{3.14}$$

$$\to$$

$$-\left\langle \frac{1}{\rho}\frac{\delta F}{\delta \theta}\nabla\theta, \frac{\delta H}{\delta \mathbf{u}}\right\rangle \qquad -L^\theta_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\Big(\theta; \Big(\frac{\delta F}{\delta \theta}\Big)_{\mathbf{u}}\Big) - \left\langle \Big(\widetilde{\widetilde{\frac{\delta F}{\delta \theta}}}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\Big(\frac{\delta H}{\delta \mathbf{u}}\Big)\Big)\right\rangle, \tag{3.15}$$

where the terms

$$\Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}, \qquad \Big(\widetilde{\widetilde{\frac{\delta F}{\delta \theta}}}\Big)_{\mathbf{u}}, \tag{3.16}$$

and all following double-tilde ones are defined analogously to $\tilde{\tilde{\gamma}}_{\mathbf{u}}$ in Definition 7. (3.15) corresponds to the right-hand side of the thermal field advection equation (3.8), while (3.14) does to the corresponding antisymmetric bracket term, which appears as a forcing term in the momentum equation. Note that L^θ is bilinear in F and H (considering its definition (1.144)) and further that since both θ^a and $\mathbb{S}_{\mathbf{u}}$ are linear, the corresponding upwind contribution bracket term is also bilinear. Since (3.14) and (3.15) form an anti-symmetric pair by construction, we therefore find that the modified, upwinded bracket given in Definition 19 below is still an antisymmetric bilinear form. In Proposition 20, we confirm that this setup does indeed lead to an SUPG equation of the form (3.6).

Definition 19. *The SUPG almost Poisson bracket is given by*

$$\{F,H\} = -\left\langle \tfrac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \tfrac{1}{\rho}\tfrac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \tfrac{\delta H}{\delta \rho}, \nabla \cdot \tfrac{\delta F}{\delta \mathbf{u}} \right\rangle + L^{\theta}_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\tfrac{\delta H}{\delta \theta}\right)_{\mathbf{u}}\right) + \left\langle \left(\widetilde{\widetilde{\tfrac{\delta H}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\tfrac{\delta F}{\delta \mathbf{u}}\big)\Big)\right\rangle$$

$$- \left\langle \tfrac{\delta F}{\delta \rho}, \nabla \cdot \tfrac{\delta H}{\delta \mathbf{u}} \right\rangle - L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\tfrac{\delta F}{\delta \theta}\right)_{\mathbf{u}}\right) - \left\langle \left(\widetilde{\widetilde{\tfrac{\delta F}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\tfrac{\delta H}{\delta \mathbf{u}}\big)\Big)\right\rangle,$$

with operators L^{θ}, θ^a and $\mathbb{S}_{\mathbf{u}}$ given by Definitions 7, 17, and 18, respectively.

Proposition 20. *Consider the space discretised almost Poisson bracket given by Definition 19. Then the thermal field transport equation based on the Poisson system* (1.53) *with this bracket is given by*

$$\left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle + L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}(\theta; \gamma_{\mathbf{u}}) = 0 \qquad\qquad \forall \gamma \in \mathbb{V}_{\theta}. \tag{3.17}$$

Proof. *We consider the Poisson system with the usual functional of the form $F = \langle \gamma, \theta \rangle$. In this case, as before all parts of the bracket except for the ones corresponding to temperature advection (i.e.* (3.15)*) are zero, and we are left with*

$$\langle \gamma, \theta_t \rangle = \frac{dF}{dt} = \{F,H\} = -L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}(\theta; \gamma_{\mathbf{u}}) - \left\langle \tilde{\tilde{\gamma}}_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\frac{\delta H}{\delta \mathbf{u}}\big)\Big)\right\rangle. \tag{3.18}$$

By definition of θ^a, the first term of the right-hand side is equal to $\left\langle \gamma, \theta^a\big(\frac{\delta H}{\delta \mathbf{u}}\big)\right\rangle$, which in turn can be expressed using $\mathbb{S}_{\mathbf{u}}$ as

$$\left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\frac{\delta H}{\delta \mathbf{u}}\big)\Big)\right\rangle. \tag{3.19}$$

We can then cancel with the upwind contribution term (i.e. the last term in (3.18)*), so that*

$$\langle \gamma, \theta_t \rangle = \left\langle \gamma, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\frac{\delta H}{\delta \mathbf{u}}\big)\Big)\right\rangle, \tag{3.20}$$

which holds for any γ. Hence, the $\mathbb{S}_{\mathbf{u}}^{-1}\theta^a$ term is point-wise equal to θ_t, and we may replace it in (3.18) *to arrive at*

$$\langle \gamma, \theta_t \rangle = -L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}(\theta; \gamma_{\mathbf{u}}) - \left\langle \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle. \tag{3.21}$$

Moving the upwind contribution term and the advection operator to the left-hand side then leads to the desired upwinded form. □

The second implication of including a θ_t term in the bracket lies in the corresponding antisymmetric term (i.e. the last term in (3.14)), given by

$$\left\langle \Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\frac{\delta F}{\delta \mathbf{u}}\big)\Big) \right\rangle. \tag{3.22}$$

The test function $\frac{\delta F}{\delta \mathbf{u}}$ appears as an argument of $\mathbb{S}_{\mathbf{u}}^{-1}\theta^a$, which we would be required to solve for every test function in the variational solver for the velocity. To avoid this, we use the adjoint of $\mathbb{S}_{\mathbf{u}}^{-1}\theta^a$, allowing us to move the operator away from the test function. This leads to a formulation as given in the proposition below.

Proposition 21. *Consider the space discretised almost Poisson bracket given by Definition 19. Then the momentum equation based on the Poisson system* (1.53) *with this bracket is given by*

$$\begin{aligned} &\langle \mathbf{w}, \mathbf{u}_t \rangle + \left\langle \mathbf{w}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{w} \right\rangle \\ &\qquad - L^{\theta}_{\mathbf{w}/\rho}\Big(\theta; \Big(\frac{\delta H}{\delta \theta}\Big)_{\mathbf{u}}\Big) - \left\langle \mathbf{s}\Big(\frac{\delta H}{\delta \theta}\Big), \mathbf{w} \right\rangle = 0 \qquad\qquad \forall \mathbf{w} \in \mathbb{V}_u, \end{aligned} \tag{3.23}$$

where $\mathbf{s}\big(\frac{\delta H}{\delta \theta}\big) \in \mathbb{V}_u$ *is given by*

$$\langle \mathbf{s}, \mathbf{v} \rangle = -L^{\theta}_{\mathbf{v}/\rho}(\theta; \sigma_{\mathbf{u}}) \qquad\qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{3.24}$$

for $\sigma_{\mathbf{u}}$ *as given by* (1.143) *in Definition 7, with* $\sigma\big(\frac{\delta H}{\delta \theta}\big) \in \mathbb{V}_\theta$ *such that*

$$\big\langle \tau, \sigma + \tilde{\tilde{\sigma}}_{\mathbf{u}} \big\rangle = \left\langle \tau, \Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}} \right\rangle \qquad\qquad \forall \tau \in \mathbb{V}_\theta. \tag{3.25}$$

Proof. *The first three terms in* (3.23) *(excluding the time derivative term) follow directly from the bracket, using* $F = \langle \mathbf{w}, \mathbf{u} \rangle$ *in the Poisson system. To obtain the last term, we evaluate the corresponding bracket term* (3.22) *using adjoint operators* $(\cdot)^*$ *as*

$$\left\langle \Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\big(\frac{\delta F}{\delta \mathbf{u}}\big)\Big) \right\rangle = \left\langle (\theta^a)^*(\mathbb{S}_{\mathbf{u}}^{-1})^*\Big(\Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}\Big), \frac{\delta F}{\delta \mathbf{u}} \right\rangle \tag{3.26}$$

$$= \left\langle (\theta^a)^*(\mathbb{S}_{\mathbf{u}}^{*})^{-1}\Big(\Big(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}\Big), \frac{\delta F}{\delta \mathbf{u}} \right\rangle, \tag{3.27}$$

noting that in the last line we swapped the adjoint and inverse operations applied on $\mathbb{S}_{\mathbf{u}}$ [*]. *The adjoints* $(\theta^a)^*$ *and* $\mathbb{S}^*_{\mathbf{u}}$ *are given in Definitions 17 and 18, respectively. Note that given a field* $\sigma \in \mathbb{V}_\theta$*, the resulting adjoint operator field* $\chi = \mathbb{S}^*_{\mathbf{u}}(\sigma)$ *is given by*

$$\langle \tau, \chi \rangle = \langle \tau, \sigma + \tilde{\tilde{\sigma}}_{\mathbf{u}} \rangle \qquad \forall \tau \in \mathbb{V}_\theta. \tag{3.28}$$

To obtain the inverse, we fix χ *in* (3.28) *and solve for* $\sigma = (\mathbb{S}^*_{\mathbf{u}})^{-1}(\chi)$*. In particular, in* (3.27)*, we need to solve this for*

$$\chi = \Big(\tilde{\tilde{\frac{\delta H}{\delta \theta}}}\Big)_{\mathbf{u}}, \tag{3.29}$$

which yields σ *as defined in* (3.25) *above. Finally, we need to compute* $(\theta^a)^*(\sigma)$*, whose solution is given by* $\mathbf{s}$ *as defined in* (3.24) *above.* □

Using Propositions 20 and 21, from the SUPG bracket in Definition 19 we obtain a system of equations of the form

$$\begin{aligned} &\langle \mathbf{w}, \mathbf{u}_t \rangle + \Big\langle \mathbf{w}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \Big\rangle - \Big\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{w} \Big\rangle \\ &\qquad - L^\theta_{\mathbf{w}/\rho}\Big(\theta; \Big(\frac{\delta H}{\delta \theta}\Big)_{\mathbf{u}}\Big) - \Big\langle \mathbf{s}\Big(\tilde{\tilde{\frac{\delta H}{\delta \theta}}}\Big), \mathbf{w} \Big\rangle = 0 \qquad \forall \mathbf{w} \in \mathbb{V}_u, \end{aligned} \tag{3.30}$$

$$\langle \phi, \rho_t \rangle + \Big\langle \phi, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \Big\rangle = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \tag{3.31}$$

$$\Big\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \Big\rangle + L^\theta_{\frac{\delta H}{\delta \mathbf{u}}/\rho}(\theta; \gamma_{\mathbf{u}}) = 0 \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{3.32}$$

Note that following Section 1.2, we find that the Poisson system can be recovered from these equations from the chain rule for F as in (1.63). Since the SUPG bracket is anti-symmetric, we therefore find that equations (3.30) - (3.32) are still energy conserving.

Remark 22. *To also incorporate energy conserving upwind methods for the advection of* $\mathbf{u}$ *and* ρ *in the thermodynamic equation set* (3.30) - (3.32)*, we can use the framework provided in the previous chapter. Note that the velocity recovery operator is applied to all test functions on the right hand side of the momentum equation* (2.34)*. For a consistent use of test functions, when extending the previous chapter's bracket to the thermal case, we then also apply this operator to the test functions of the corresponding thermal bracket*

[*] We have $\iota = \iota^* = \big((\mathbb{S}^*_{\mathbf{u}})^{-1}\mathbb{S}^*_{\mathbf{u}}\big)^*$, for identity operator ι. But $\big((\mathbb{S}^*_{\mathbf{u}})^{-1}\mathbb{S}^*_{\mathbf{u}}\big)^* = (\mathbb{S}^*_{\mathbf{u}})^*\big((\mathbb{S}^*_{\mathbf{u}})^{-1}\big)^* = \mathbb{S}_{\mathbf{u}}\big((\mathbb{S}^*_{\mathbf{u}})^{-1}\big)^*$. Hence, $(\mathbb{S}_{\mathbf{u}})^{-1} = \big((\mathbb{S}^*_{\mathbf{u}})^{-1}\big)^*$, and applying $(\cdot)^*$ again we obtain $((\mathbb{S}_{\mathbf{u}})^{-1})^* = (\mathbb{S}^*_{\mathbf{u}})^{-1}$.

term (3.14)*. To maintain antisymmetry, we then also need to apply it in* (3.15)*, and we arrive at a fully upwinded bracket as given in Definition 9, together with*

$$
\begin{aligned}
&+ L^{\theta}_{\mathbb{U}\left(\rho, \frac{\delta F}{\delta \mathbf{u}}\right)}\left(\theta; \left(\frac{\delta H}{\delta \theta}\right)_{\mathbf{u}}\right) + \left\langle \left(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\mathbb{U}(\rho, \frac{\delta F}{\delta \mathbf{u}})\right)\right)\right\rangle \\
&- L^{\theta}_{\mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right)}\left(\theta; \left(\frac{\delta F}{\delta \theta}\right)_{\mathbf{u}}\right) - \left\langle \left(\widetilde{\widetilde{\frac{\delta F}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\mathbb{U}(\rho, \frac{\delta H}{\delta \mathbf{u}})\right)\right)\right\rangle,
\end{aligned}
\tag{3.33}
$$

and θ^a *now defined by*

$$
\langle \gamma, \theta^a(\mathbf{v}) \rangle = -L^{\theta}_{\mathbf{v}}(\theta; \gamma_{\mathbf{u}}) \qquad\qquad \forall \gamma \in \mathbb{V}_{\theta}. \tag{3.34}
$$

The flux recovered velocity $\mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right)$ *then also serves as the advecting velocity for the thermal field transport equation, given by*

$$
\left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle + L^{\theta}_{\mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right)}(\theta; \gamma_{\mathbf{u}}) \qquad\qquad \forall \gamma \in \mathbb{V}_{\theta}. \tag{3.35}
$$

Finally, we note that the energy conserving 2D velocity advection and Coriolis terms in (2.30) *and* (2.31) *can be extended readily to the three-dimensional case* (1.132)*, according to*

$$
\begin{aligned}
&\left\langle \nabla \times \left(\rho \mathbb{U}(\rho, \frac{\delta F}{\delta \mathbf{u}}) \times \mathbb{U}(\rho, \frac{\delta H}{\delta \mathbf{u}})\right), \mathbf{u} \right\rangle - \int_{\Gamma} \{\{\mathbf{n} \times \left(\rho \mathbb{U}(\rho, \frac{\delta F}{\delta \mathbf{u}}) \times \mathbb{U}(\rho, \frac{\delta H}{\delta \mathbf{u}})\right)\}\} \cdot \tilde{\mathbf{u}} \, dS \\
&- \left\langle \rho \mathbb{U}(\rho, \frac{\delta F}{\delta \mathbf{u}}), 2\boldsymbol{\Omega} \times \mathbb{U}(\rho, \frac{\delta H}{\delta \mathbf{u}}) \right\rangle.
\end{aligned}
\tag{3.36}
$$

The resulting bracket then leads to a fully upwinded energy conserving space discretisation for the 3D compressible Euler equations.

3.2 Time discretisation

In this section, we first present the fully energy conserving scheme, and describe the resulting non-linear set of equations. We then discuss its operational applicability, and introduce an alternative scheme which is fully upwinded, leads to a reduced computational cost, and is energy conserving up to a small approximation in the time discretisation.

3.2.1 Fully energy conserving formulation and nonlinear system

To confirm the energy conserving property of the space discretised equations (3.30) - (3.32), we follow the approach in the previous chapter and apply the energy conserving Poisson integrator (1.116), thus expecting energy conservation up to solver tolerance (see Section (1.2.4)). The resulting nonlinear system of equations is given by

$$\Big\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n \Big\rangle + \Delta t \Bigg(\Big\langle \mathbf{w}, \bar{\boldsymbol{\omega}} \times \frac{1}{\bar{\rho}} \overline{\frac{\delta H}{\delta \mathbf{u}}} \Big\rangle - \Big\langle \overline{\frac{\delta H}{\delta \rho}}, \nabla \cdot \mathbf{w} \Big\rangle$$

$$- L^{\theta}_{\mathbf{w}/\bar{\rho}} \Big(\bar{\theta}; \Big(\overline{\frac{\delta H}{\delta \theta}} \Big)_{\bar{\mathbf{u}}} \Big) - \Big\langle \mathbf{s} \Big(\overline{\overline{\frac{\delta H}{\delta \theta}}} \Big), \mathbf{w} \Big\rangle \Bigg) = 0 \quad \forall \mathbf{w} \in \mathbb{V}_u, \quad (3.37)$$

$$\Big\langle \phi, \rho^{n+1} - \rho^n \Big\rangle + \Delta t \Big\langle \phi, \nabla \cdot \overline{\frac{\delta H}{\delta \mathbf{u}}} \Big\rangle = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \quad (3.38)$$

$$\Big\langle \gamma_{\bar{\mathbf{u}}}, \theta^{n+1} - \theta^n \Big\rangle + \Delta t L^{\theta}_{\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}} (\bar{\theta}; \gamma_{\bar{\mathbf{u}}}) = 0 \qquad \forall \gamma \in \mathbb{V}_\theta, \quad (3.39)$$

for midpoint time average $\bar{\rho} = (\rho^n + \rho^{n+1})/2$, and similar for $\bar{\mathbf{u}}$ and $\bar{\boldsymbol{\omega}} = (\bar{\rho}\bar{q})\mathbf{k}$, with q^n and q^{n+1} given by the diagnostic scalar vorticity equation (3.5), using the prognostic fields at time levels n and $n+1$, respectively. Note that all occurrences of the state $\mathbf{z}$ in L^θ and $\mathbf{s}$ are also discretised at the half-time level. Further, we remark that the derivation of the fully discretised thermal field equation (3.39) follows the proof of Proposition 20 in a time discretised form, with θ_t replaced by $(\theta^{n+1} - \theta^n)/\Delta t$.

To solve for $\mathbf{z}^{n+1}$, given the fully discretised residual $\mathbf{R}(\mathbf{z}^{n+1})$ above (left-hand sides of (3.37) - (3.39)), we apply the same procedure as detailed in the previous chapter and revert to a Picard iteration method. For this purpose, we recall that for update $\delta\mathbf{z} = \mathbf{z}^{n+1,k+1} - \mathbf{z}^{n+1,k}$ with unknown next time step estimate $\mathbf{z}^{n+1,k+1}$, we set

$$\frac{\delta \mathbf{R}'}{\delta \mathbf{z}}(\delta \mathbf{z}) = -\mathbf{R}(\mathbf{z}^{n+1,k}), \quad (3.40)$$

and $\mathbf{z}^{n+1,0} = \mathbf{z}^n$. The left hand side corresponds to the Jacobian of a linearised version of $\mathbf{R}$ without Hamiltonian projections, and for the Euler equations it is given by

$$\frac{\delta \mathbf{R}'}{\delta \mathbf{z}}(\delta \mathbf{z}) = \begin{pmatrix} \langle \delta \mathbf{u}, \mathbf{w} \rangle + \frac{\Delta t}{2} \Big(\langle 2\boldsymbol{\Omega} \times \delta \mathbf{u}, \mathbf{w} \rangle - \Big\langle g\delta\rho + c_p(\bar{\bar{\theta}}\delta\pi + \delta\theta\bar{\bar{\pi}}), \nabla \cdot \mathbf{w} \Big\rangle - c_p \langle \bar{\bar{\pi}} \nabla \delta\theta, \mathbf{w} \rangle \Big) \\ \langle \delta\rho, \phi \rangle + \frac{\Delta t}{2} \langle \bar{\bar{\rho}} \nabla \cdot \delta \mathbf{u}, \phi \rangle \\ \langle \delta\theta, \gamma \rangle \end{pmatrix}, \tag{3.41}$$

where double-barred entries correspond to background fields, with $\bar{\bar{\pi}} = \pi(\bar{\bar{\rho}}, \bar{\bar{\theta}})$, $\bar{\bar{\mathbf{u}}} = \mathbf{0}$, and $\delta\pi = \frac{\partial \pi}{\partial \rho}(\bar{\bar{\rho}}, \bar{\bar{\theta}})\delta\rho + \frac{\partial \pi}{\partial \theta}(\bar{\bar{\rho}}, \bar{\bar{\theta}})\delta\theta$. Note that in view of the test cases to follow, we assumed the background thermal field to be constant (equal to g for the thermal shallow water equations, and an isentropic background potential temperature for the Euler equations). This leads to a vanishing $\nabla\bar{\bar{\theta}}$ term in the velocity and thermal field equation, respectively, thus uncoupling the latter from the density and velocity equations. Given the right-hand side $-\mathbf{R}(\mathbf{z}^{n+1,k})$, we can then first solve for $\delta\theta$, followed by a mixed solve for the density and velocity updates. Note that since we focus on the underlying space discretisation, the nonlinear solve procedure is kept simple. The discretisation can in principle be applied equally to fully three-dimensional problems and scenarios with a non-constant potential background temperature (by approximately eliminating $\delta\theta$ in the linearised momentum equation). A suitable nonlinear solver strategy for this case can be found in [BGS+19].

3.2.2 Fully upwinded, approximately energy conserving formulation

While both the space and time discretisations conserve energy, the resulting Picard iteration method does not. In the numerical tests below, we find that the number of Picard iterations required to achieve energy conservation up to machine precision is much higher than is usually considered in forecasting models. This suggests that we may simplify the fully discretised scheme such that the additional energy error is smaller than the error due to a small number of Picard iterations. Further, the SUPG bracket presented in Definition 19 does not include upwinding in the velocity and density fields, leading to a non-satisfactory quality of the field development. In this section, we present a fully upwinded fully discretised formulation based on the bracket as described in Remark 22 and a small modification to the Poisson time discretisation.

In the fully upwinded bracket, we find that in the temperature advection equation the advecting and upwind velocities are given by $\mathbb{U}(\rho, \frac{\delta H}{\delta \mathbf{u}})$ and $\mathbf{u}$, respectively, which time-discretise within the Poisson integrator as $\overline{\frac{\delta H}{\delta \mathbf{u}}}$ (given by (1.123)) and $\bar{\mathbf{u}}$, respectively. The equations derived from the bracket can be simplified by ensuring that these two veloc-

ities are equal after time-discretisation, which can be achieved by time discretising the Hamiltonian variation in $\mathbf{u}$ as

$$\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_u}(\rho \mathbf{u}) \quad \rightarrow \quad P_{\mathbb{V}_u}(\bar{\rho}\bar{\mathbf{u}}) =: \frac{\overline{\delta H}'}{\delta \mathbf{u}}\,. \tag{3.42}$$

In this case, as mentioned in Remark 13 in the previous chapter, we find that the velocity recovery operator $\mathbb{U}$ is evaluated as

$$\left\langle \bar{\rho}\mathbf{v}, \mathbb{U}\left(\bar{\rho}, \frac{\overline{\delta H}'}{\delta \mathbf{u}}\right)\right\rangle = \langle \mathbf{v}, P_{\mathbb{V}_u}(\bar{\rho}\bar{\mathbf{u}})\rangle = \langle \mathbf{v}, \bar{\rho}\bar{\mathbf{u}}\rangle \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{3.43}$$

so that

$$\mathbb{U}\left(\bar{\rho}, \frac{\overline{\delta H}'}{\delta \mathbf{u}}\right) = \bar{\mathbf{u}}, \tag{3.44}$$

point-wise. With this in mind, we can replace the last terms in the thermal field part (3.33) of the fully upwinded bracket by

$$+\left\langle \Big(\frac{\widetilde{\widetilde{\delta H}}}{\delta\theta}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\Big(\mathbb{U}\big(\rho, \frac{\delta F}{\delta \mathbf{u}}\big)\Big)\Big)\right\rangle \qquad + \left\langle \Big(\frac{\widetilde{\widetilde{\delta H}}}{\delta\theta}\Big)_{\mathbb{U}\left(\rho, \frac{\delta F}{\delta \mathbf{u}}\right)}, \mathbb{S}_{\mathbf{u}}^{-1}\big(\theta^a(\mathbf{u})\big)\right\rangle \tag{3.45}$$

$$\rightarrow$$

$$-\left\langle \Big(\frac{\widetilde{\widetilde{\delta F}}}{\delta\theta}\Big)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\Big(\theta^a\Big(\mathbb{U}\big(\rho, \frac{\delta H}{\delta \mathbf{u}}\big)\Big)\Big)\right\rangle \qquad - \left\langle \Big(\frac{\widetilde{\widetilde{\delta F}}}{\delta\theta}\Big)_{\mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right)}, \mathbb{S}_{\mathbf{u}}^{-1}\big(\theta^a(\mathbf{u})\big)\right\rangle, \tag{3.46}$$

that is we swapped the advecting and upwind velocities. Note that the new pair of terms is still bilinear and antisymmetric (given the upwind definition (1.142)), and in particular, this would not be the case if we simply replaced $\mathbb{U}\big(\rho, \frac{\delta F}{\delta \mathbf{u}}\big)$ and $\mathbb{U}\big(\rho, \frac{\delta H}{\delta \mathbf{u}}\big)$ by $\mathbf{u}$. In the full Poisson integrator setup with this modification of the bracket, the proof of Proposition 20 does not work anymore, since the advection operator term (3.19) does not cancel with the new upwind contribution term (3.46). However, the cancellation (and therefore the proof) still works for the time-discretised form (3.42) of the Hamiltonian variation in $\mathbf{u}$, since in this case the advecting and upwind velocities are equal.

Crucially, after swapping the velocities, we find in (3.45) that the double operator $\mathbb{S}_{\mathbf{u}}^{-1}\theta^a$ is now applied to $\mathbf{u}$ instead of a test function $\frac{\delta F}{\delta \mathbf{u}}$, so that the adjoint formulation described in Proposition 21 is not needed anymore.

Proposition 23. *Consider the fully upwinded Poisson bracket as defined in Remark 22, together with modification* (3.45) - (3.46)*. Then applying the Poisson time discretisation*

(1.116) *with modification* (3.42) *leads to a fully discretised set of equations of the form*

$$\Big\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n \Big\rangle - \Delta t \Big(\big\langle \nabla^\perp \big(\bar{\rho} \mathbb{U}(\bar{\rho}, \mathbf{w}) \cdot \bar{\mathbf{u}}^\perp \big), \bar{\mathbf{u}} \big\rangle - \int_\Gamma [\![\bar{\rho} \mathbb{U}(\bar{\rho}, \mathbf{w}) \cdot \bar{\mathbf{u}}^\perp]\!] \mathbf{n}^\perp \cdot \tilde{\mathbf{u}} \, dS \Big)$$

$$+ \Delta t \Big(\big\langle \bar{\rho} \mathbb{U}(\bar{\rho}, \mathbf{w}), f \bar{\mathbf{u}}^\perp \big\rangle + \Big\langle \bar{\rho} \, \mathbb{U}(\bar{\rho}, \mathbf{w}), \nabla \overline{\frac{\delta H}{\delta \rho}} \Big\rangle - \int_\Gamma [\![\overline{\frac{\delta H}{\delta \rho}} \mathbb{U}(\bar{\rho}, \mathbf{w})]\!] \tilde{\bar{\rho}} \, dS \Big)$$

$$- \Delta t L^\theta_{\mathbb{U}(\bar{\rho}, \mathbf{w})} \Big(\bar{\theta}; \big(\overline{\frac{\delta H}{\delta \theta}} \big)_{\bar{\mathbf{u}}} \Big) - \Big\langle \big(\overline{\overline{\frac{\delta H}{\delta \theta}}} \big)_{\mathbb{U}(\bar{\rho}, \mathbf{w})}, \theta^{n+1} - \theta^n \Big\rangle = 0 \qquad \forall \mathbf{w} \in \mathbb{V}_u, \quad (3.47)$$

$$\big\langle \phi, \rho^{n+1} - \rho^n \big\rangle - \Delta t \Big(\langle \nabla \phi, \bar{\rho} \bar{\mathbf{u}} \rangle - \int_\Gamma [\![\phi \bar{\mathbf{u}}]\!] \tilde{\bar{\rho}} \, dS \Big) = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \quad (3.48)$$

$$\big\langle \gamma + \tilde{\bar{\gamma}}_{\bar{\mathbf{u}}}, \theta^{n+1} - \theta^n \big\rangle + \Delta t L^\theta_{\bar{\mathbf{u}}}(\bar{\theta}; \gamma_{\bar{\mathbf{u}}}) = 0 \qquad \forall \gamma \in \mathbb{V}_\theta. \quad (3.49)$$

Proof. *All terms follow directly from the fully upwinded bracket together with the advecting velocity* (3.44)*, except for the ones related to the modification* (3.45) - (3.46)*. For the temperature SUPG related term* (3.46)*, an argument identical to the one used in Proposition 20 can be used to arrive at the final form* (3.49) *of the thermal field equation. It remains to resolve the antisymmetric forcing term* (3.45)*. Using definition* (3.34) *of* θ^a *together with the definition* (3.11) *of* $\mathbb{S}$*, we can rewrite the thermal field equation* (3.49) *as a point-wise relation of the form*

$$\frac{\theta^{n+1} - \theta^n}{\Delta t} = \mathbb{S}^{-1}_{\bar{\mathbf{u}}} \big(\theta^a(\bar{\mathbf{u}}) \big). \quad (3.50)$$

Replacing the corresponding term in (3.45)*, we arrive at the last term in* (3.47)*.* □

Remark 24. *To solve for the velocity contribution* (3.47) *within the right-hand side* $\mathbf{R}$ *of the Picard iteration method* (3.40)*, we need to take care of the modified test functions* $\mathbb{U}(\bar{\rho}, \mathbf{w})$*. For this purpose, we apply the procedure as described in Remark 12 of the previous chapter.*

Note that as described in Remark 22, the above set of equations can be adjusted to three-dimensional scenarios by using velocity advection and Coriolis terms of the form (3.36). Further, note that since the underlying bracket is still antisymmetric, the equation set formulated in Proposition 23 is energy conserving up to the time discretisation. The only difference to the fully energy conserving Poisson integrator lies in the simplified variation

of the Hamiltonian in $\mathbf{u}$, and comparing it to the non-simplified one, we find that

$$\frac{\overline{\delta H}}{\delta \mathbf{u}} - \frac{\overline{\delta H}'}{\delta \mathbf{u}} = \frac{1}{3} P_{\mathbb{V}_u}(\rho^n \mathbf{u}^n + \frac{1}{2}\rho^n \mathbf{u}^{n+1} + \frac{1}{2}\rho^{n+1}\mathbf{u}^n + \rho^{n+1}\mathbf{u}^{n+1}) - P_{\mathbb{V}_u}(\bar{\rho}\bar{\mathbf{u}}) \tag{3.51}$$

$$= \frac{1}{12} P_{\mathbb{V}_u}\big((\rho^{n+1} - \rho^n)(\mathbf{u}^{n+1} - \mathbf{u}^n)\big) \tag{3.52}$$

$$= \frac{1}{12} P_{\mathbb{V}_u}\big((\rho'^{\,n+1} - \rho'^{\,n})(\mathbf{u}'^{\,n+1} - \mathbf{u}'^{\,n})\big), \tag{3.53}$$

where $'$ denotes perturbation from the reference state. We therefore expect this modification to only have a small effect on the energy development, especially when only a small number of Picard iterations is used, so that the energy error is dictated by the error due to the nonlinear solver strategy.

Finally, compared to the equations derived from the bracket as presented in Definition 19, the equation set as given in Proposition 23 is fully upwinded and only requires additional projections for the Hamiltonian variations in ρ and θ. The variation in ρ is a projection into a DG space, while the variation in θ is a projection into a horizontally discontinuous space in the case of the Euler equations, implying that the additional cost in evaluating the residual in each Picard iteration is small compared to similar discretisations derived in a non-Hamiltonian, non-energy conserving context.

3.3 Numerical results

In this section, we confirm numerically the upwind-stabilised and energy conserving properties of the fully discretised schemes (3.37) - (3.39) and (3.47) - (3.49). For comparison purposes, we first describe two non-energy conserving space discretisations. We then consider test cases consisting of a perturbed thermogeostrophic balance scenario for the thermal shallow water equations, as well as a cold and hot air bubble scenario for the Euler equations.

3.3.1 Comparison to non-energy conserving space discretisations

The first non-energy conserving space discretisation is formulated such that the Poisson time integrator can be applied to it, allowing for a comparison that focuses on the exact energy conserving properties of the SUPG stabilised space discretisation. The second is formulated using a standard treatment of the Euler equations, and is used in comparison with the fully upwinded scheme as described in Section 3.2.2, with a smaller number of

Picard iterations in the simulation runs.

For the first discretisation, we use a non-antisymmetric bracket of the form

$$-\left\langle \tfrac{\delta F}{\delta \mathbf{u}}, \mathbf{q} \times \tfrac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \tfrac{\delta H}{\delta \rho}, \nabla \cdot \tfrac{\delta F}{\delta \mathbf{u}} \right\rangle - \left\langle \nabla \cdot \left(\tfrac{1}{\rho} \tfrac{\delta H}{\delta \theta} \tfrac{\delta F}{\delta \mathbf{u}} \right), \theta \right\rangle + \textstyle\int_\Gamma [\![\tfrac{1}{\rho} \tfrac{\delta H}{\delta \theta} \tfrac{\delta F}{\delta \mathbf{u}}]\!] \mathrm{avg}(\theta)\, dS \tag{3.54}$$

$$- \left\langle \tfrac{\delta F}{\delta \rho}, \nabla \cdot \tfrac{\delta H}{\delta \mathbf{u}} \right\rangle - L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho} \left(\theta; \left(\tfrac{\delta F}{\delta \theta} \right)_{\mathbf{u}} \right) - \left\langle \left(\widetilde{\tfrac{\delta F}{\delta \theta}} \right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1} \left(\theta^a \left(\tfrac{\delta H}{\delta \mathbf{u}} \right) \right) \right\rangle, \tag{3.55}$$

recalling that avg$(\cdot)$ denotes the average across facets. Note that this bracket is antisymmetric with respect to the velocity and density advection terms, but not with respect to the thermal field advection term. The last two terms in (3.55) are set to arrive at the SUPG stabilised form (3.17) of the thermal field advection equation, while the last two terms of (3.54) correspond to a simple stabilisation of the non-upwinded thermal field bracket term:

$$\left\langle \frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla \theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \quad \rightarrow \quad - \left\langle \nabla \cdot \left(\frac{1}{\rho} \frac{\delta H}{\delta \theta} \frac{\delta F}{\delta \mathbf{u}} \right), \theta \right\rangle + \int_\Gamma \left[\!\!\left[\frac{1}{\rho} \frac{\delta H}{\delta \theta} \frac{\delta F}{\delta \mathbf{u}} \right]\!\!\right] \mathrm{avg}(\theta)\, dS. \tag{3.56}$$

This allows us to investigate the loss in energy due to a non-energy conserving implementation of the SUPG method.

The second discretisation is derived directly from the continuous Euler equations (1.98) - (1.104), disregarding the Hamiltonian framework. For this purpose, we use the same types of upwinding as discussed in Section 1.3, i.e. standard DG upwinding for ρ, SUPG for θ, and Lie derivative type upwinding the rotational part of velocity advection. The resulting weak form is then given by

$$\langle \mathbf{w}, \mathbf{u}_t \rangle - \left\langle \nabla^\perp \left(\mathbf{w} \cdot \mathbf{u}^\perp \right), \mathbf{u} \right\rangle + \int_\Gamma [\![\mathbf{w} \cdot \mathbf{u}^\perp]\!] \mathbf{n}^\perp \cdot \tilde{\mathbf{u}}\, dS - \left\langle \nabla \cdot \mathbf{w}, |\mathbf{u}|^2 \right\rangle + \langle \mathbf{w}, g\mathbf{k} \rangle$$

$$- c_p \langle \nabla \cdot (\theta \mathbf{w}), \pi \rangle + c_p \int_\Gamma [\![\theta \mathbf{w}]\!] \{\pi\}\, dS = 0 \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{3.57}$$

$$\langle \phi, \rho_t \rangle - \langle \nabla \phi, \rho \mathbf{u} \rangle + \int_\Gamma [\![\phi \mathbf{u}]\!] \tilde{\rho}\, dS = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \tag{3.58}$$

$$\left\langle \gamma + \tilde{\tilde{\gamma}}_{\mathbf{u}}, \theta_t \right\rangle + L^{\theta}_{\mathbf{u}}(\theta; \gamma_{\mathbf{u}}) = 0 \qquad \forall \theta \in \mathbb{V}_\theta, \tag{3.59}$$

noting that for the Euler vertical slice test cases considered below, the Coriolis parameter f is set zero. We use midpoint averages for the time discretisation, so that the resulting fully discretised density and thermal field equations are equal to the ones following from

the approximately energy conserving discretisation given in Proposition 23. In particular, in comparison to this non-Hamiltonian discretisation, we find that the adjustments used to derive the approximately energy conserving discretisation lie entirely in the velocity equation, with a forcing contribution given by the last two lines in (3.47) and a modified test function of the form $\mathbb{U}(\bar{\rho}, \mathbf{w})$.

3.3.2 Test cases

Having formulated the fully energy conserving scheme (3.37) - (3.39), as well as the approximately energy conserving scheme (Proposition 23) and two non-energy conserving schemes, we test for their energy-conserving properties and the qualitative field development. We first consider a thermal shallow water test case and an Euler test case, demonstrating energy conservation up to solver tolerance as well as an improved thermal field development when the SUPG method is applied. Given the shortcomings of the field development in the Euler test case, we then move on to the fully upwinded setup, which we investigate using two Euler test cases.

As in the previous chapter, the test cases were implemented using the automated finite element toolkit Firedrake (see near the bottom of Section 2.2.1). For the thermal shallow water test case, we consider a set of compatible finite element spaces given by $(P_3,\ BDM_2,\ P_1^{DG})$, together with P_3 for the buoyancy field. For the Euler test cases, we consider $(CG_k,\ H(\mathrm{div})_k,\ DG_{k-1})$ with $k = 2$ unless states otherwise, together with the corresponding Charney-Phillips type potential temperature space. Further, for simplicity, we set the SUPG parameter to $\tau = \Delta t/2$ for all test cases (see Appendix C.3.1).

Energy conservation and thermal field upwinding

The thermal shallow water test case considered here is based on the fifth test case in [WDH+92], corresponding to a spherical flow in geostrophic balance, perturbed by a mountain. Following [EDK19], we set the buoyancy field such that the overall state is in thermogeostrophic balance. The initial conditions are then given by

$$\mathbf{u} = u_0(-y, x, 0)/a, \tag{3.60}$$

$$\rho = \bar{\bar{\rho}} - (a\Omega u_0 + u_0^2/2)\frac{z^2}{ga^2} - b, \tag{3.61}$$

$$\theta = g\left(1 + \epsilon\left(\frac{\bar{\bar{\rho}}}{\rho}\right)^2\right), \tag{3.62}$$

where as in the previous chapter, $b = b_0(1 - r/R)$ describes the mountain's surface, which is defined as before. $\bar{\bar{\rho}}$ denotes the mean height, which, like all other parameters, is also given as before. Finally, the additional parameter in the buoyancy field is given by $\epsilon = 0.05$.

The mesh is given by an icosahedral triangulation, where as in the previous chapter refinement level 0 corresponds to 20 triangles. For every higher level, each triangle is refined to 4 triangles (so that each increase corresponds to halving the cell side length Δx), and in this test case the refinement level was set to 4. The simulation is run for 50 days, with a time step of $\Delta t = 8$ minutes, and 8 Picard iterations for each time step.

The resulting buoyancy fields for the fully energy conserving discretised equations corresponding to the non-upwinded bracket (3.1) and the SUPG bracket (Definition 19) are depicted in Figure 3.1. Further, the relative energy error (i.e. $(E_t - E_0)/E_0$) development for the aforementioned brackets as well as the non-energy conserving one ((3.54) - (3.55)) are depicted in Figure 3.2.

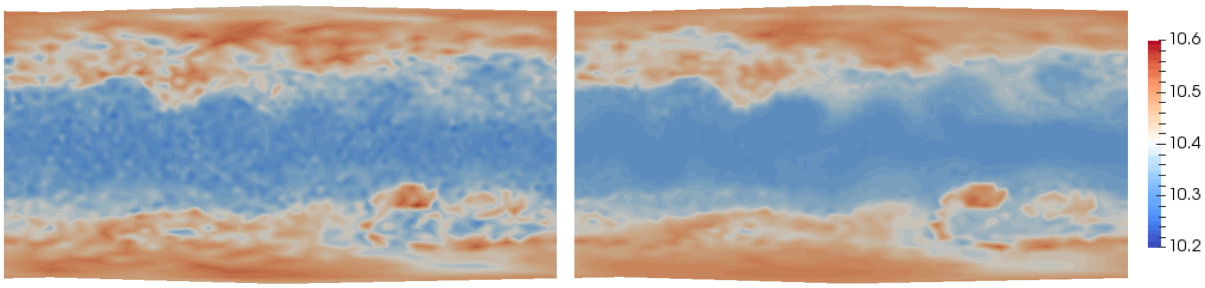


Figure 3.1: Buoyancy fields for thermal Williamson 5 test case after 50 days, for energy conserving discretisations. Left: no SUPG for buoyancy field; right: SUPG for buoyancy field. Mesh refinement level 4, $\Delta t = 8$ minutes, with 8 Picard iterations per time step.

As expected, we find that the incorporation of SUPG markedly reduces the occurrence of spurious oscillations in the buoyancy field development. Further, both energy conserving brackets lead to energy convergence up to solver tolerance, with an improvement by 4 orders of magnitude when compared to the non-energy conserving bracket.

Next, we consider the brackets for the Euler equations. Since the non-upwinded bracket (3.1) leads to an unstable field development in the test case described below, we instead compare the SUPG bracket to one that also includes DG-upwinding for potential temper-

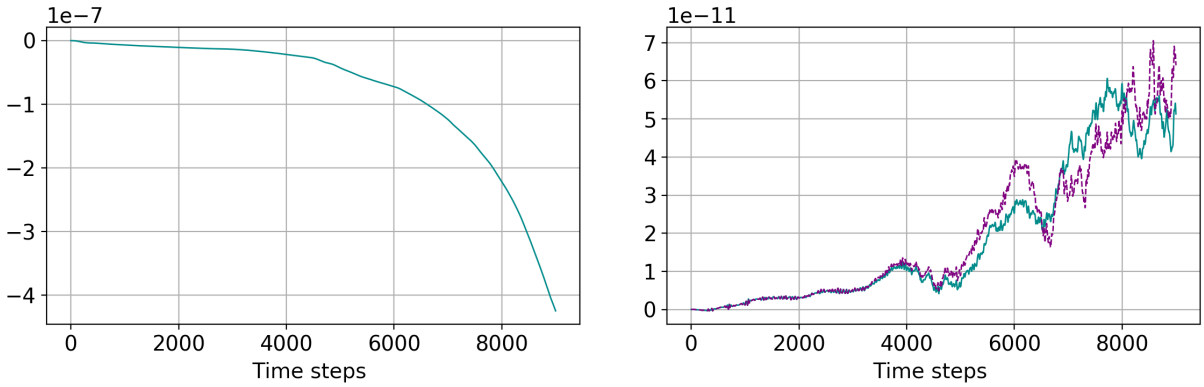


Figure 3.2: Relative energy error development for thermal Williamson 5 test case. Left: non-energy conserving bracket. Right: energy conserving bracket with SUPG for buoyancy (cyan) and energy conserving bracket without SUPG for buoyancy (dashed purple).

ature in the horizontal direction, but no SUPG in the vertical. Note that such a bracket can be derived from the SUPG bracket by setting the SUPG parameter τ to zero. Further, note that the diagnostic vorticity equation (3.5) does not conserve vorticity near the boundary even in the absence of a baroclinic term, which may lead to an unbounded growth of enstrophy due to sources near the boundary (for more details, see [BC18] and Appendix D). However, for the test case below this effect is small, and given our focus on the temperature advection method we ignore it.

The test case is given by a falling bubble in a vertical slice of the atmosphere, based on one as described in [SWW+93]. For this purpose, we consider a horizontally periodic rectangular domain Ω of 32 km length and 6.4 km height, with a constant potential temperature background field $\bar{\bar{\theta}} = 300$ K and corresponding pressure and density fields in hydrostatic balance. A temperature perturbation of the form

$$\Delta T = \begin{cases} -\frac{15}{2}(1+\cos(r\pi)) & \text{if } r < 1, \\ 0 & \text{otherwise,} \end{cases} \qquad r = \sqrt{\frac{(x-x_c)^2}{x_r^2} + \frac{(z-z_c)^2}{z_r^2}}, \tag{3.63}$$

is added to the background potential temperature, while the density field is left unperturbed. The perturbation's horizontal and vertical centre and radius are given by $(x_c, x_r) = (16, 4)$ and $(z_c, z_r) = (3, 2)$ kilometres, respectively. The gravitational acceleration is defined as in the thermal shallow water test case, and the remaining physical parameters are given by $c_v = 716.5$ m^{2}s^{-2}K^{-1}, $R = 287$ m^{2}s^{-2}K^{-1}, and $p_0 = 100$ kPa. Finally, we note that a constant viscosity term is commonly added to the continuous equations in this test case to obtain a solution that converges as the resolution is refined [MDW+10]. In our case, we do not include this term due to our focus on energy conser-

vation.

The mesh is given by a vertically extruded interval mesh, with horizontal and vertical resolutions equal to $\Delta x = \Delta z = 100$ m. The simulation is run for 900 seconds, with a time step of $\Delta t = 0.5$ s, and 32 Picard iterations for each time step.

To focus on the effects of upwinding, we consider the field development after 400 and 800 seconds. The resulting images and relative energy error developments are depicted in Figures 3.3 and 3.4, in an arrangement equal to the one for the thermal shallow water equations above.

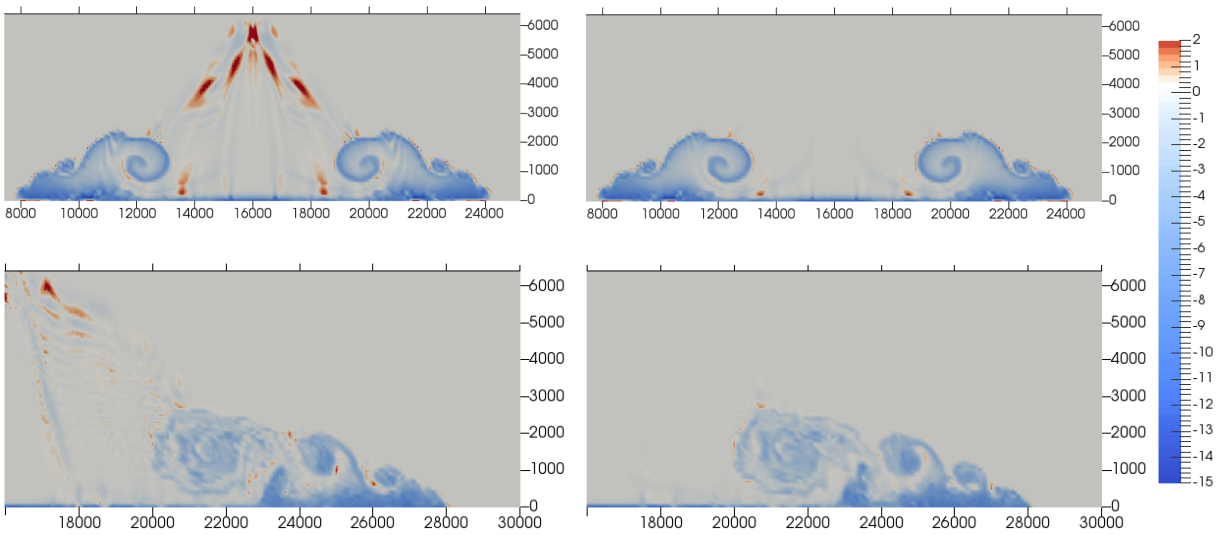


Figure 3.3: Potential temperature fields for falling bubble test case after 400 s (top row) and 800 s (bottom row), for energy conserving discretisations. Left column: no SUPG for potential temperature field; right column: SUPG for potential temperature field. $\Delta x = \Delta z = 100$ m, $\Delta t = 0.5$ s, with 32 Picard iterations per time step.

We find that if the SUPG method is not applied in the vertical direction of potential temperature advection, spurious upward moving features occur in the potential temperature field development near the centre of the domain, where the bubble falls towards the bottom boundary. In the presence of the SUPG method these features are removed, again indicating a qualitatively favourable field development if the method is included. Further, as expected, energy is conserved up to solver tolerance for the two energy conserving brackets, with occasional jumps likely due to an insufficient number of Picard iterations. In contrast, the non-energy conserving bracket leads to a loss of energy of the order of 10^{-6}, 6 orders of magnitude larger than the energy conserving ones.

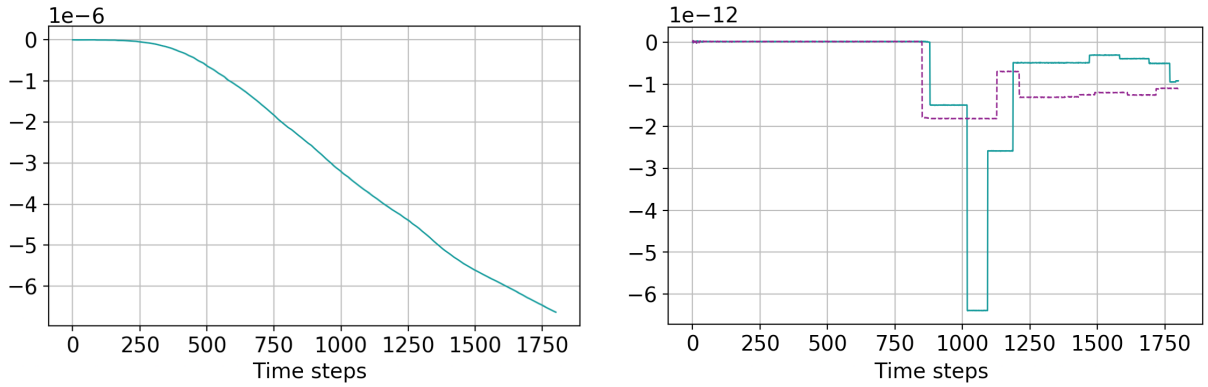


Figure 3.4: Relative energy error development for falling bubble test case. Left: non-energy conserving bracket. Right: energy conserving bracket with SUPG for buoyancy (cyan) and energy conserving bracket without SUPG for buoyancy (dashed purple).

Fully upwinded, approximately energy conserving scheme

While including the SUPG method for potential temperature advection leads to an improvement of the field development, the latter still suffers from an absence of upwinding in the other fields, leading to an insufficient resolution of the density current flowing along the bottom boundary in the Euler test case. Further, 32 Picard iterations were required to achieve energy conservation of the order of 10^{-12}. We therefore next consider the field and energy error developments for the approximately energy conserving scheme as presented in Proposition 23, which is energy conserving in space and energy conserving in time up to the difference given by (3.53). The test cases considered here are given by the falling bubble one as described above, as well as a rising bubble test case based on [CJDWH90]. The latter is prone to secondary plumes (see e.g. [BGS+19]), and to avoid these in our discussion here, we consider the next higher order set of finite element spaces ($k = 3$) for this test case.

The domain Ω is given by a horizontally periodic square shaped vertical slice of the atmosphere of 10 km side length. The background fields are given as in the falling bubble case, and the initial potential temperature perturbation is given by

$$\Delta\theta = \begin{cases} \Delta\theta_{mag} \cos^2(\frac{r\pi}{2}) & \text{if } r < 1, \\ 0 & \text{otherwise,} \end{cases} \tag{3.64}$$

for r as in (3.63) above, with $(x_c, x_r) = (5, 2)$ and $(z_c, z_r) = (2, 2)$ kilometres, and $\Delta\theta_{mag} = 2$ K. As before, the density field is left unperturbed. For the relative energy error study below, the underlying extruded mesh is as for the falling bubble test case

with $\Delta x = \Delta z = 100$ m. For the field development figures below, a higher resolution of $\Delta x = \Delta z = 50$ m is used. The simulations are run for 1000 seconds, with a time step of $\Delta t = 1$ s for the 100 m mesh and $\Delta t = 0.5$ s for the 50 m mesh.

First, we compare the impact of both the reduced number of Picard iterations and the approximated Poisson integrator (i.e. (3.42)) on the relative energy error development. For this purpose, we run the falling and rising bubble test cases for $k \in \{4, 5, 6, 8\}$ Picard iterations, each time for both the fully energy conserving Poisson integrator and the approximated version as given in Proposition 23. The resulting energy developments are depicted in Figure 3.5.

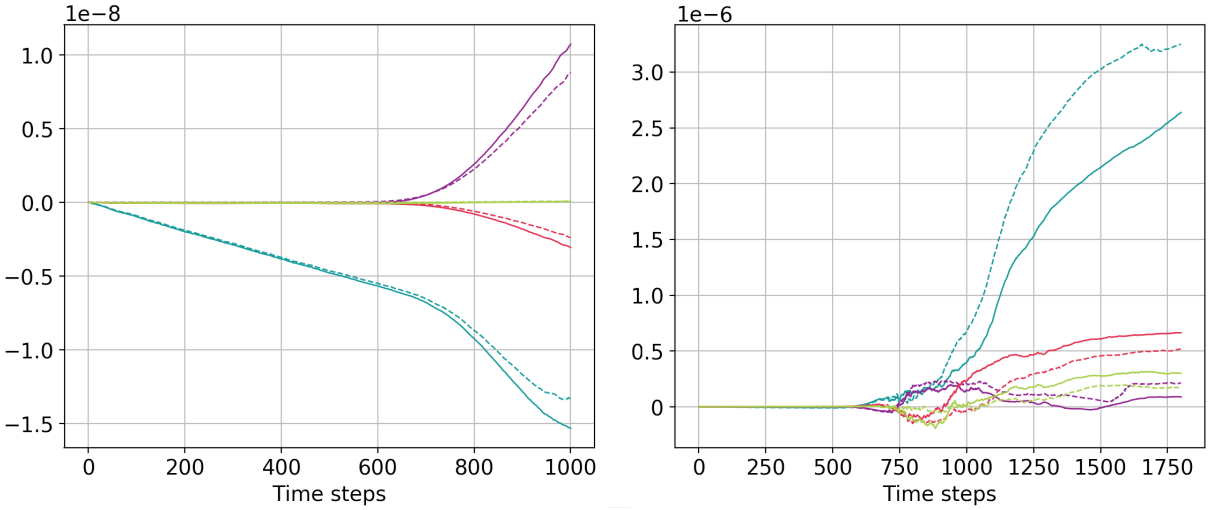


Figure 3.5: Relative energy error development for rising (left) and falling (right) bubble test cases. Solid lines indicate the approximated Poisson integrator, dashed ones the full Poisson integrator. Colours correspond to cyan for 4 Picard iterations, purple for 5, red for 6, and light green for 8.

While more evident for the less turbulent rising bubble test case, we find that in both cases the choice of number of Picard iterations dominates the relative energy error development when compared to whether or not the Poisson integrator has been approximated. In particular, this indicates that the approximated scheme as given in Proposition 23 does not lead to a substantial increase in energy error when a small number of Picard iterations is used, suggesting that the approximation (and therefore the removal of the additional computations for $\mathbb{U}$, $(\mathbb{S}_{\mathbf{u}}^{-1})^*$, and $(\theta^a)^*$ appearing in the fully upwinded, fully energy conserving discretisation) is justified.

To test the improvement of the qualitative field development when a fully upwinded bracket is used, we compare the scheme as given in Proposition 23 with a fully upwinded

reference scheme derived in a non-Hamiltonian setup (i.e. directly from the continuous equations), as defined in (3.57) - (3.59). We also consider the energy development for this comparison, using 4 Picard iterations in each time step. Note that considering the two discretisations, we find that the only additional computational cost of the approximately energy conserving scheme lies in computing $\frac{\delta H}{\delta \rho}$ and $\frac{\delta H}{\delta \theta}$ for each Picard iteration, i.e. a DG space and a horizontally DG space projection. The resulting falling and rising bubble potential temperature fields are depicted in Figure 3.6, and the corresponding energy plots can be found in Figure 3.7.

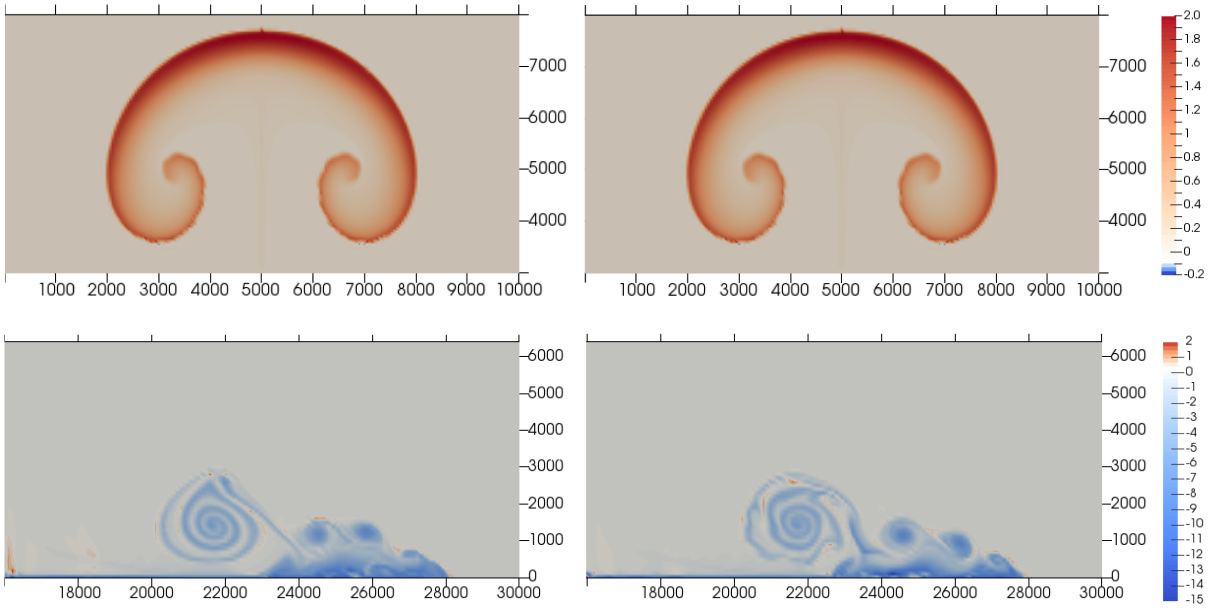


Figure 3.6: Potential temperature fields for approximately energy conserving scheme (right column) and non-energy conserving scheme (left column). Top: rising bubble test case at 1000 s. Bottom: falling bubble test case at 800 s.

For the rising bubble test case, we find a virtually identical field development. In terms of the relative energy error, we can divide the bubble's evolution into an initial phase, where the bubble starts rising, and a later phase, in which two vortices are formed at the bubble's lower ends. For the former phase, the approximately energy conserving setup suffers from a small, constant decrease in energy, probably related to the initial adjustment of the unperturbed density field to a perturbed state (in the sense that if a corresponding perturbation to the density field is added to the initial conditions, this decrease disappears). Once the vortices start to form, however, the non-energy conserving setup suffers from a much larger increase in relative energy error than the approximately energy conserving one does, leading to a final energy error larger by almost an order of magnitude.

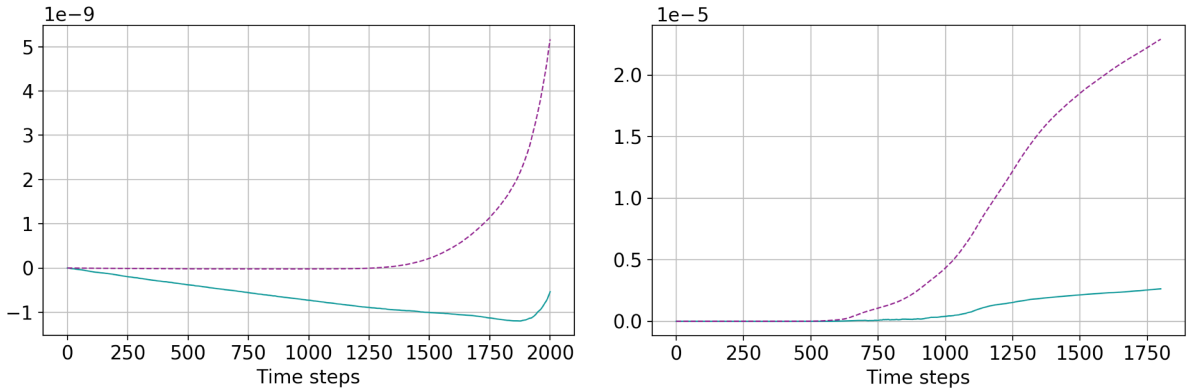


Figure 3.7: Relative energy error development for approximately energy conserving scheme (cyan) and non-energy conserving scheme (dashed purple), for 4 Picard iterations per time step. Left: rising bubble test case. Right: falling bubble test case.

Similarly, we find an improved energy conservation by nearly an order of magnitude for the falling bubble test case. Additionally, we observe a qualitatively different field development, where the higher degree of energy conservation in time leads to the formation of additional smaller vortices in the main vortex. This demonstrates that even in simulations with a comparable computational effort, the approximately energy conserving discretisation leads to a significant reduction in the relative energy error in comparison to a corresponding non-Hamiltonian discretisation, which in turn may influence the field development. Should a further reduction be required – albeit at a higher computational cost – this can readily be achieved in the approximately energy conserving setup by using a moderately higher number of Picard iterations (with another order of magnitude gained for a total of 6 iterations per time step, as indicated in Figure 3.5).

Remark 25. *The additional smaller vortices in the approximately energy conserving discretisation* (3.47) - (3.49) *(see Figure 3.6, bottom right) also occur in [MNG15] (see Figures 4 to 7), where a dynamic sub-grid scale model as described in Section 1.3.4 is used to stabilise a higher order spectral element discretisation. In our case, these features are a direct consequence of the reduction in energy conservation error, which in turn follows from the antisymmetric structure of the underlying almost Poisson bracket. The antisymmetry ensures correct transfers between kinetic, potential, and internal energy, in the sense that the growths and decreases of these three types of energy will not match exactly if the bracket is not antisymmetric. Indeed, as we move from the bracket that underlies the approximately energy conserving setup* (3.47) - (3.49) *term by term towards the non-antisymmetric formulation* (3.57) - (3.59)*, the features become less pronounced and eventually disappear (figures not shown here).*

3.4 Summary

In this chapter, we presented an almost Poisson bracket discretisation which includes an SUPG method for the thermal field advection. The bracket was used to derive fully energy conserving schemes for the Euler and thermal rotating shallow water equations. The SUPG formulation relies on moving the upwinded part of the test function corresponding to the thermal field time derivative into the Poisson bracket. In view of the Poisson time integrator, a temperature advection and an SUPG related operator were introduced in order to reformulate the thermal field time derivative term occurring in the bracket. We demonstrated how to recover the thermal and velocity field evolution equations given the SUPG-modified bracket, showing that the thermal field equation corresponds to the usual SUPG formulation of a transport equation with the given underlying finite element spaces. Further, for the interest of a fully upwinded, computationally less costly discretisation, we introduced an approximately energy conserving version including upwinding in the density and velocity fields.

In numerical tests, we demonstrated energy conservation up to solver tolerance for the SUPG-modified bracket for an Euler and a thermal rotating shallow water scenario. Further, the incorporation of the SUPG method was shown to lead to a significant improvement of the qualitative thermal field development in both cases. Finally, we considered the approximately energy conserving discretisation in a simulation with a small number of Picard iterations, showing that we still gain a reduced energy error when compared to a reference discretisation derived in a non-Hamiltonian setup, at a comparable computational cost. For the more turbulent of the two Euler test cases considered here, the reduced energy error additionally leads to a qualitatively different field development when compared to the reference discretisation. It arises in the form of additional small scale features, which resemble those of an alternative spectral element discretisation found in the literature, where a dynamic sub-grid scale model is used. In future work, we aim to investigate the origin of these features in the energy conserving framework. In particular, this includes a study of the energy transfers related to the Poisson bracket's antisymmetric structure.

4

Vorticity stabilisation

As noted in Remark 16 in Chapter 2, the rotating shallow water equations also conserve enstrophy. In numerical schemes, it is desirable to dissipate the latter since it cascades to and eventually accumulates at small scales. Unlike the velocity upwind method described in Section 1.3.2, the approach in [BC18], where the SUPG method is applied to the vorticity evolution equation, allows for a more direct control of the dissipation of enstrophy. Additionally, it was shown in [NC17a] that for sets of lowest (polynomial) order compatible finite element spaces, the vorticity SUPG approach leads to improved results when compared to the one based on velocity upwinding. For these reasons, and noting that the UK Met Office's next dynamical core is based on a lowest order set of compatible spaces, it is desirable to extend the approach in [BC18] to the thermal shallow water and three-dimensional compressible Euler equations.

In this chapter, we describe the incorporation of the SUPG method for vorticity stabilisation into the standard bracket (1.57), and by extension into the thermal field SUPG bracket given in Definition 19. Further, we present numerical results by means of the thermal rotating shallow water as well as 2D and 3D compressible Euler equations. For this purpose, we extend the framework of [BC18], where such a stabilisation method is described for the rotating shallow water equations in the presence of boundaries, using a bracket akin to the standard one. The remainder of this chapter is organised as follows: in Section 4.1, we describe the extension to the thermal as well as three-dimensional case. We also include a remark on the applicability of this framework in the case of magne-

tohydrodynamics, since the vorticity and magnetic field evolution equations are closely related. In Section 4.2, we discuss technical details of the vorticity SUPG formulation with respect to the energy conserving Poisson time integrator. In Section 4.3, we present numerical results, confirming energy conservation and an improved field development due to vorticity SUPG when compared to velocity upwinding. Finally, in Section 4.4, we summarise the chapter. The extension of the boundary treatment is detailed in Appendix D.

4.1 SUPG stabilisation in the discretised bracket

To set the scene, we briefly review the framework established in [BC18] for the rotating shallow water case, with a discretised almost Poisson bracket given by

$$\{F, H\} = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, q \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle. \tag{4.1}$$

Note that in order to focus on the incorporation of the SUPG method, we consider a domain without boundaries; for details on the case with boundaries, see Appendix D. Further, in view of the extension to follow, we use a density field related variable ρ, rather than the more common depth field notation D. From the Poisson system (1.53) together with this bracket, we obtain a momentum equation given by

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = -\left\langle \mathbf{w}, q \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{w} \right\rangle \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{4.2}$$

where we further recall from Section (1.2) that $\frac{\delta H}{\delta \mathbf{u}}$ is equal to the flux $P_{\mathbb{V}_u}(\rho \mathbf{u})$, for projection $P_{\mathbb{V}_u}$, and similarly for $\frac{\delta H}{\delta \rho}$. Given the diagnostic vorticity equation (1.112),

$$\langle \eta, q\rho \rangle = -\langle \nabla^{\perp} \eta, \mathbf{u} \rangle + \langle \eta, f \rangle \qquad \forall \eta \in \mathbb{V}_q, \tag{4.3}$$

we then further obtain, upon differentiating (4.3) with respect to time and inserting into the momentum equation (4.2), a vorticity evolution equation of the form

$$\langle \eta, (q\rho)_t \rangle = -\langle \nabla^{\perp} \eta, \mathbf{u}_t \rangle = \left\langle \nabla^{\perp} \eta, q \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle - \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \nabla^{\perp} \eta \right\rangle = \left\langle \nabla \eta, q \frac{\delta H}{\delta \mathbf{u}} \right\rangle, \tag{4.4}$$

$$\implies \left\langle \eta, (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle = 0 \qquad \forall \eta \in \mathbb{V}_q, \tag{4.5}$$

where we used $\nabla \cdot \nabla^{\perp}\eta = 0$, and integration by parts for the remaining term. Note that since η is continuous, and $\frac{\delta H}{\delta \mathbf{u}} \in \mathbb{V}_u$, the resulting facet integral vanishes. In practice, when considering the fully discretised scheme, we then first solve for (4.5) in each nonlinear iteration of a given time step, and use the resulting vorticity field in the momentum equation (4.2); a description of this process is given at (4.96) - (4.98) in Section 4.2.1 below.

Conservation of enstrophy, i.e. $q^2\rho$, can be demonstrated by setting $\eta = q$, which yields

$$\int_\Omega (q^2\rho)_t \, dx = \langle q, (q\rho)_t \rangle + \langle q_t, q\rho \rangle = \langle q, 2(q\rho)_t \rangle - \langle q^2, \rho_t \rangle \tag{4.6}$$

$$= -2 \left\langle q, \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle + \left\langle q^2, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle \tag{4.7}$$

$$= \left\langle \nabla q^2, \frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle q^2, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle = \int_\Omega \nabla \cdot \left(q^2 \frac{\delta H}{\delta \mathbf{u}} \right) \, dx = 0, \tag{4.8}$$

noting that we used the L^2-equivalence of ρ_t and $-\nabla \cdot \frac{\delta H}{\delta \mathbf{u}}$. The latter follows from the continuity equation derived using the underlying bracket (4.1), and we recall that it holds since both ρ_t and $\nabla \cdot \frac{\delta H}{\delta \mathbf{u}}$ are functions of the depth space $\mathbb{V}_\rho$. Further, we used integration by parts, and as above, since q is continuous and $\frac{\delta H}{\delta \mathbf{u}} \in \mathbb{V}_u$, the facet integral contributions vanish.

To incorporate the SUPG stabilisation method into the weak form (4.5), we would normally proceed as in Section 1.3.3 and modify the test functions according to

$$\eta \to \eta + \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla \eta. \tag{4.9}$$

The use of the flux based form $\frac{\delta H}{\delta \mathbf{u}}/\rho = P_{\mathbb{V}_u}(\rho \mathbf{u})/\rho$, rather than simply $\mathbf{u}$, will become apparent in the enstrophy derivation (4.15) below. However, since the vorticity equation was derived from the momentum equation, we need to ensure, for the purpose of consistency, that all SUPG related modifications also occur in the latter. To guide this process, we consider the SUPG method in its residual based form (1.151), that is

$$\left\langle \eta, (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle = - \left\langle \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla \eta, q_{res} \right\rangle \qquad \forall \eta \in \mathbb{V}_q, \tag{4.10}$$

where the residual corresponds to the vorticity evolution equation's strong form, which follows directly from the weak equation (4.5), i.e.

$$(q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) = 0. \tag{4.11}$$

In view of the incorporation of vorticity SUPG into the almost Poisson bracket framework as described in Proposition 26 below, we then set

$$q_{res} = (q\rho)_t + \nabla \cdot (q\mathbf{F}), \tag{4.12}$$

for $\mathbf{F} = P_{\mathbb{V}_u}(\rho\mathbf{u})$. Note that after discretisation, q_{res} need not be zero. Further, note that while $\frac{\delta H}{\delta \mathbf{u}}$ is also equal to $P_{\mathbb{V}_u}(\rho\mathbf{u})$, from a Poisson system point of view $\mathbf{F}$ will be considered as a quantity that appears as part of the bracket, while $\frac{\delta H}{\delta \mathbf{u}}$ is a quantity derived from the Hamiltonian (see Remark 27 and Section 4.2 below). Finally, the strong vorticity evolution equation (4.11) is given in flux-conservative form, which is specific to the two-dimensional case. It is also possible to specify the residual in advective form according to

$$q_{res} = \rho \left(q_t + \mathbf{F}/\rho \cdot \nabla q \right), \tag{4.13}$$

which is derived from (4.12) using the (strong) continuity equation $\rho_t + \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} = 0$ together with $\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_q}(\rho\mathbf{u}) = \mathbf{F}$.

To illustrate that the resulting SUPG stabilised equation (4.10) together with the residual (4.12) dissipates enstrophy, we revisit (4.8), where the $(q\rho)_t$ related term in line (4.6) is now specified according to (4.10), rather than the non-stabilised form (4.5). The rate of change of total enstrophy is then given by

$$\int_\Omega (q^2\rho)_t \, dx = -2 \left\langle \tau \frac{\delta H}{\delta \mathbf{u}}/\rho \cdot \nabla \eta, q_{res} \right\rangle \tag{4.14}$$

$$= -2 \left(\| \sqrt{\tau/\rho} \, P_{\mathbb{V}_q}(\rho\mathbf{u}) \cdot \nabla q \|_2^2 + \langle \tau P_{\mathbb{V}_q}(\rho\mathbf{u}) \cdot \nabla q, q_t \rangle \right), \tag{4.15}$$

where the second line readily follows using the residual's advective form (4.13), and where we again relied on $\frac{\delta H}{\delta \mathbf{u}} = P_{\mathbb{V}_q}(\rho\mathbf{u}) = \mathbf{F}$. As in the thermal case in Chapter 3, the first term on the right-hand side acts as dissipation along streamlines, while the sign-indefinite second one is kept small by a suitable choice of τ.

Crucially, the fact that the right-hand side in the residual based form (4.10) can be

rewritten according to

$$-\left\langle \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla \eta, q_{res} \right\rangle = -\left\langle \tau \frac{\delta H}{\delta \mathbf{u}}^{\perp} / \rho \cdot \nabla^{\perp} \eta, q_{res} \right\rangle, \tag{4.16}$$

then allows us to incorporate the SUPG stabilisation method for vorticity within the momentum equation.

Proposition 26. *([BC18]) Consider a discretised almost Poisson bracket defined by*

$$\{F, H\} = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, q^* \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle, \tag{4.17}$$

for

$$q^* = q - \frac{\tau}{\rho} q_{res}, \qquad q_{res} = (q\rho)_t + \nabla \cdot (q\mathbf{F}), \qquad \mathbf{F} = P_{\mathbb{V}_u}(\rho \mathbf{u}), \tag{4.18}$$

where $P_{\mathbb{V}_u}$ denotes projection into $\mathbb{V}_u$. Then the momentum equation derived from the corresponding Poisson system is given by

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = -\left\langle \mathbf{w}, q^* \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \mathbf{w} \right\rangle \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{4.19}$$

which admits an SUPG stabilised vorticity evolution equation of the form

$$\left\langle \eta, (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle = -\left\langle \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla \eta, q_{res} \right\rangle \qquad \forall \eta \in \mathbb{V}_q. \tag{4.20}$$

Proof. *The momentum equation follows directly from the Poisson system. To obtain the vorticity evolution equation, we follow the derivation of the weak form* (4.5)*, and arrive at*

$$\langle \eta, (q\rho)_t \rangle = -\langle \nabla^{\perp} \eta, \mathbf{u}_t \rangle = \left\langle \nabla^{\perp} \eta, q^* \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle = \left\langle \nabla \eta, q^* \frac{\delta H}{\delta \mathbf{u}} \right\rangle = \left\langle \nabla \eta, \left(q - \frac{\tau}{\rho} q_{res} \right) \frac{\delta H}{\delta \mathbf{u}} \right\rangle \tag{4.21}$$

$$= -\left\langle \eta, \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle - \left\langle \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla \eta, q_{res} \right\rangle \qquad \forall \eta \in \mathbb{V}_q. \tag{4.22}$$

Moving the advection term (first term on the right-hand side in (4.22)*) to the left-hand side then yields the required form. Finally, as shown in* (4.15)*, the additional τ term acts as SUPG stabilisation.* □

Remark 27. *Note that* $\mathbf{F}$ *in* q_{res} *is defined as* $P_{\mathbb{V}_u}(\rho \mathbf{u})$ *and not* $\frac{\delta H}{\delta \mathbf{u}}$*, since otherwise the*

resulting bracket's velocity advection related term

$$-\left\langle \frac{\delta F}{\delta \mathbf{u}}, \left(q - \frac{\tau}{\rho}\left((q\rho)_t + \nabla\cdot\left(q\frac{\delta H}{\delta \mathbf{u}}\right)\right)\right)\frac{\delta H}{\delta \mathbf{u}}^{\perp}\right\rangle \tag{4.23}$$

would no longer be bilinear in F and H. Further, note that as seen in Section 3.2.2 in the previous chapter, $\mathbf{F}$ and $\frac{\delta H}{\delta \mathbf{u}}$ are equal only up the choice of time discretisation, and in particular they differ if a Poisson integrator is used. We will elaborate on this difference more in Section 4.2 below.

Next to facilitating the incorporation of vorticity SUPG into the momentum equation, the reformulation of SUPG stabilisation according to the residual form (4.10) also points to a possible generalisation towards the thermal and three-dimensional cases. In the next two sections, we first focus on the thermal extension, followed by the extension to three dimensions.

4.1.1 Thermal two-dimensional case

To include a thermal field related variable, we extend the shallow water bracket (4.1) analogously to the standard one (1.57), that is we add an antisymmetric pair of terms

$$+\left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}}\right\rangle - \left\langle \frac{1}{\rho}\frac{\delta F}{\delta \theta}\nabla\theta, \frac{\delta H}{\delta \mathbf{u}}\right\rangle. \tag{4.24}$$

In this case, we obtain a momentum equation given by

$$\langle \mathbf{w}, \mathbf{u}_t\rangle = -\left\langle \mathbf{w}, q\frac{\delta H}{\delta \mathbf{F}}^{\perp}\right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla\cdot\mathbf{w}\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \mathbf{w}\right\rangle \qquad \forall \mathbf{w}\in\mathbb{V}_u, \tag{4.25}$$

and the additional baroclinic forcing term (i.e. the last one in (4.25)) then also appears in the vorticity equation, which now reads

$$\langle \eta, (q\rho)_t\rangle = -\left\langle \eta, \nabla\cdot\left(q\frac{\delta H}{\delta \mathbf{u}}\right)\right\rangle - \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla^{\perp}\eta\right\rangle \qquad \forall \eta\in\mathbb{V}_q. \tag{4.26}$$

Note that in the thermal case, enstrophy is no longer conserved, and repeating the derivation (4.8), we find

$$\int_\Omega (q^2\rho)_t\, dx = \langle q, 2(q\rho)_t\rangle - \langle q^2, \rho_t\rangle = -2\left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla^{\perp}q\right\rangle. \tag{4.27}$$

To include SUPG stabilisation, we proceed as before and consider a residual based extension to (4.26), i.e.

$$\left\langle \eta, (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) \right\rangle + \left\langle \frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta, \nabla^\perp \eta \right\rangle = - \left\langle \tau \frac{\delta H}{\delta \mathbf{u}} / \rho \cdot \nabla\eta, q^\theta_{res} \right\rangle \quad \forall \eta \in \mathbb{V}_q, \tag{4.28}$$

where q^θ_{res} corresponds to the vorticity equation's strong form with a baroclinic forcing term. To derive this form, we consider (4.26) before discretisation, with suitably differentiable solution spaces $\mathbb{V}_q$, $\mathbb{V}_\theta$, et cetera. Then applying integration by parts for $\nabla^\perp$ in the baroclinic term, we arrive at

$$\left\langle \eta, (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) - \nabla^\perp \cdot \left(\frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta \right) \right\rangle = 0 \qquad \forall \eta \in \mathbb{V}_q, \tag{4.29}$$

$$\implies \qquad (q\rho)_t + \nabla \cdot \left(q \frac{\delta H}{\delta \mathbf{u}} \right) - \nabla^\perp \cdot \left(\frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta \right) = 0, \tag{4.30}$$

and set q^θ_{res} equal to the left-hand side of (4.30).

Remark 28. *To illustrate the additional forcing term, we can consider e.g. the compressible Euler equations. Following Section 1.2.3, we find that in this case* $\frac{\delta H}{\delta \theta} = c_p \rho \pi$*, so that*

$$\nabla^\perp \cdot \left(\frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta \right) = c_p \nabla^\perp \pi \cdot \nabla\theta, \tag{4.31}$$

where we used the product rule and $\nabla^\perp \cdot \nabla\theta = 0$*. This is the two-dimensional version of the more well-known baroclinic term* $c_p \nabla\pi \times \nabla\theta$*, which demonstrates that the strength of baroclinic forcing depends on how "non-aligned" the pressure and temperature gradients are (noting that* $\mathbf{a} \times \mathbf{b} = 0$ *for parallel vectors* $\mathbf{a}$*,* $\mathbf{b}$*).*

From here, an SUPG stabilisation for the weak discretised vorticity equation can be obtained in an equal manner to the non-thermal case.

Proposition 29. *Consider a discretised almost Poisson bracket defined by*

$$\begin{aligned} \{F, H\} = &- \left\langle \frac{\delta F}{\delta \mathbf{u}}, q^* \frac{\delta H}{\delta \mathbf{u}}^\perp \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \\ &- \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{1}{\rho} \frac{\delta F}{\delta \theta} \nabla\theta, \frac{\delta H}{\delta \mathbf{u}} \right\rangle, \end{aligned} \tag{4.32}$$

for

$$q^* = q - \frac{\tau}{\rho} q^\theta_{res}, \qquad q^\theta_{res} = (q\rho)_t + \nabla \cdot (q\mathbf{F}) - \nabla^\perp \cdot (T/\rho \nabla\theta), \tag{4.33}$$

where $\mathbf{F}$ *and* T *are set equal to the projections resulting from the Hamiltonian variations*

in $\mathbf{u}$ *and* θ*, respectively, that is*

$$\mathbf{F} = P_{\mathbb{V}_u}(\rho\mathbf{u}), \qquad T = \begin{cases} P_{\mathbb{V}_\theta}\big(\rho(\frac{\rho}{2}+b)\big) & \text{thermal shallow water equations,} \\ P_{\mathbb{V}_\theta}\big(c_p\rho\pi(\rho,\theta)\big) & \text{Euler equations.} \end{cases} \tag{4.34}$$

Then the momentum equation derived from the corresponding Poisson system is given by

$$\langle \mathbf{w}, \mathbf{u}_t\rangle = -\left\langle \mathbf{w}, q^* \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla\cdot\mathbf{w} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \mathbf{w}\right\rangle \qquad \forall \mathbf{w}\in\mathbb{V}_u, \tag{4.35}$$

which admits an SUPG stabilised vorticity evolution equation of the form

$$\left\langle \eta, (q\rho)_t + \nabla\cdot\left(q\frac{\delta H}{\delta\mathbf{u}}\right)\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \nabla^\perp\eta\right\rangle = -\left\langle \tau\frac{\delta H}{\delta\mathbf{u}}/\rho\cdot\nabla\eta, q^\theta_{res}\right\rangle \quad \forall\eta\in\mathbb{V}_q. \tag{4.36}$$

Proof. *The proof is analogous to the shallow water case in Proposition 26. Dissipation of enstrophy also follows analogously to* (4.15) *(up to the baroclinic forcing term), and we find*

$$\int_\Omega (q^2\rho)_t\, dx = -2\left(\left\langle \tau\frac{\delta H}{\delta\mathbf{u}}/\rho\cdot\nabla q, q^\theta_{res}\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \nabla^\perp q\right\rangle\right) \tag{4.37}$$

$$= -2\Bigg(\|\sqrt{\tau/\rho}\, P_{\mathbb{V}_u}(\rho\mathbf{u})\cdot\nabla q\|_2^2 + \langle \tau P_{\mathbb{V}_u}(\rho\mathbf{u})\cdot\nabla q, q_t\rangle \tag{4.38}$$

$$- \left\langle \tau P_{\mathbb{V}_u}(\rho\mathbf{u})/\rho\cdot\nabla q, \nabla^\perp\cdot(T/\rho\nabla\theta)\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \nabla^\perp q\right\rangle\Bigg), \tag{4.39}$$

where we used $\frac{\delta H}{\delta\mathbf{u}} = P_{\mathbb{V}_q}(\rho\mathbf{u}) = \mathbf{F}$*. The first term in* (4.38) *ensures dissipation, while the last term in* (4.38) *and first term in* (4.39) *are kept small by a suitable choice of* τ*.* □

Note that like $\mathbf{F}$, the projection T is set equal to the one resulting from the variation of the Hamiltonian H with respect to θ, since otherwise the almost Poisson bracket would not be bilinear (see Remark 27).

Remark 30. *Considering the vorticity SUPG stabilised brackets* (4.17) *and* (4.32)*, we find that the modification required to achieve this stabilisation is restricted to the bracket term corresponding to velocity advection, i.e.*

$$-\left\langle \frac{\delta F}{\delta\mathbf{u}}, q\frac{\delta H}{\delta\mathbf{u}}^{\perp}\right\rangle \quad\longrightarrow\quad -\left\langle \frac{\delta F}{\delta\mathbf{u}}, q^*\frac{\delta H}{\delta\mathbf{u}}^{\perp}\right\rangle. \tag{4.40}$$

In particular, it is straightforward to apply this modification to almost Poisson brackets

that also include stabilisation methods for other fields, such as the previous chapter's thermal SUPG bracket given in Definition 19.

As a caveat, it should be noted that next to the SUPG modification $q \to q^*$ used to stabilise the vorticity advection term, we also still need to be able to derive a vorticity evolution equation in the first place. This is not the case – at least not in a straightforward manner – for the brackets introduced in the previous two chapters which include the velocity recovery operator $\mathbb{U}$. To illustrate this, we recall the space discretised momentum equation (2.34) *considered in Chapter 2, which is given by*

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = \Big(L_{\mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right)}\big(\mathbf{u}; \rho \mathbb{U}(\rho, \mathbf{w})\big) \tag{4.41}$$

$$+ F_{\left(\rho, \mathbb{U}\left(\rho, \frac{\delta H}{\delta \mathbf{u}}\right), \frac{\delta H}{\delta \rho}\right)}\big(\mathbb{U}(\rho, \mathbf{w})\big)\Big) =: G\big(\mathbb{U}(\rho, \mathbf{w})\big) \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{4.42}$$

where L and F correspond to advection and forcing operators, respectively. To treat $\mathbb{U}$ applied to test functions, we can solve for $\mathbf{u}^\dagger$ such that

$$\langle \rho \mathbf{v}, \mathbf{u}^\dagger \rangle = G(\mathbf{v}) \qquad \forall \mathbf{v} \in \mathbb{V}_u, \tag{4.43}$$

$$\implies \langle \mathbf{w}, \mathbf{u}_t \rangle = G\big(\mathbb{U}(\rho, \mathbf{w})\big) = \langle \rho \mathbb{U}(\rho, \mathbf{w}), \mathbf{u}^\dagger \rangle = \langle \mathbf{w}, \mathbf{u}^\dagger \rangle, \qquad \forall \mathbf{w} \in \mathbb{V}_u. \tag{4.44}$$

From the point of view of deriving a vorticity evolution equation, we would proceed as before by differentiating the diagnostic vorticity equation in time and substituting for the momentum equation. The resulting equation is then given by

$$\langle \eta, (q\rho)_t \rangle = -G\big(\mathbb{U}(\rho, \nabla^\perp \eta)\big) \qquad \forall \eta \in \mathbb{V}_q. \tag{4.45}$$

In this case, however, we cannot follow the approach (4.43) *used in the velocity equation, since the corresponding auxiliary field $q^\dagger$ defined by*

$$\langle \rho \eta, q^\dagger \rangle = G(\nabla^\perp \eta) \qquad \forall \eta \in \mathbb{V}_q, \tag{4.46}$$

does not have an obvious analogue to (4.44)*, since $\mathbb{U}$ is an operator defined on the velocity space, not the vorticity space.*

As a workaround, we can instead use the modified DG upwind method presented in Section 2.1.1, which does not require the velocity recovery operator $\mathbb{U}$. This way, we arrive at a

fully upwind-stabilised bracket given by

$$\begin{aligned}\{F,H\} = &-\left\langle \tfrac{\delta F}{\delta \mathbf{u}}, q^* \tfrac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle - \left\langle \tfrac{\delta F}{\delta \mathbf{u}}, \nabla \tfrac{\delta H}{\delta D} \right\rangle + \textstyle\int_\Gamma \left[\!\left[\tfrac{\delta H}{\delta D} \right]\!\right] \operatorname{avg}\left(\tfrac{\delta F}{\delta \mathbf{u}}/D \right) \tilde{D}\, dS \\ &+ \left\langle \tfrac{\delta H}{\delta \mathbf{u}}, \nabla \tfrac{\delta F}{\delta D} \right\rangle - \textstyle\int_\Gamma \left[\!\left[\tfrac{\delta F}{\delta D} \right]\!\right] \operatorname{avg}\left(\tfrac{\delta H}{\delta \mathbf{u}}/D \right) \tilde{D}\, dS \\ &+ L^{\theta}_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\tfrac{\delta H}{\delta \theta}\right)_{\mathbf{u}}\right) + \left\langle \left(\widetilde{\widetilde{\tfrac{\delta H}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\tfrac{\delta F}{\delta \mathbf{u}}\right)\right) \right\rangle \\ &- L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\tfrac{\delta F}{\delta \theta}\right)_{\mathbf{u}}\right) - \left\langle \left(\widetilde{\widetilde{\tfrac{\delta F}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\tfrac{\delta H}{\delta \mathbf{u}}\right)\right) \right\rangle,\end{aligned} \tag{4.47}$$

where the last two lines follow from the SUPG method applied to the thermal field, as described in Section 3.1. Note that the above discussion equally applies to the three-dimensional extension discussed below. Further, we note that for the test cases considered in the numerical results section below, we do not include the aforementioned modified DG upwind method, since there is no occurrence of strong gradients in the density related field. Instead, for simplicity we consider brackets of the form

$$\begin{aligned}\{F,H\} = \{F,H\}_u &+ \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle \\ &+ L^{\theta}_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\frac{\delta H}{\delta \theta}\right)_{\mathbf{u}}\right) + \left\langle \left(\widetilde{\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\frac{\delta F}{\delta \mathbf{u}}\right)\right) \right\rangle \\ &- L^{\theta}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\frac{\delta F}{\delta \theta}\right)_{\mathbf{u}}\right) - \left\langle \left(\widetilde{\widetilde{\widetilde{\frac{\delta F}{\delta \theta}}}}\right)_{\mathbf{u}}, \mathbb{S}_{\mathbf{u}}^{-1}\left(\theta^a\left(\frac{\delta H}{\delta \mathbf{u}}\right)\right) \right\rangle,\end{aligned} \tag{4.48}$$

where the velocity advection related term $\{F,H\}_u$ is set to the vorticity SUPG methods as described in Proposition 29 above, i.e.

$$\{F,H\}_u = -\left\langle \frac{\delta F}{\delta \mathbf{u}}, q^* \frac{\delta H}{\delta \mathbf{u}}^{\perp} \right\rangle, \tag{4.49}$$

and Proposition 31 below, as well as the velocity upwind method as given in (2.27) *(including its 3D version).*

4.1.2 Thermal three-dimensional case

The importance of enstrophy as a quadratic invariant arises due to the absence of the vortex stretching term $(\boldsymbol{\omega} \cdot \nabla \mathbf{u})$ in two dimensions (for more details, see e.g. [Val17]). In

three dimensions, this no longer applies, and we could instead aim to dissipate another relevant vorticity related field. One is given by the helicity $K = \langle \mathbf{u}, \boldsymbol{\omega} \rangle$ [AK99], which is an invariant in the case of constant potential temperature [Mor98]. However, K, which is an inner product of a velocity and a vorticity space element, is difficult to approach after discretisation. Another candidate could be Ertel's potential vorticity of the form $\nabla\theta \cdot \boldsymbol{\omega}/\rho$ [Val17], which like potential temperature is conserved along the flow. However, similarly to helicity, this type of vorticity related field also consists of a product of elements of different spaces. For simplicity, we therefore instead aim to dissipate the total amount of absolute vorticity $\boldsymbol{\omega}$. To further justify this, note that when using a vorticity evolution equation, the absolute vorticity appears in an advection operator, which may cause spurious oscillations that may be mitigated by means of dissipation, as discussed in Appendix C.

For this purpose, we consider the standard bracket (1.57), given by

$$\begin{aligned} \{F, H\} \coloneqq &- \left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho} \frac{\delta H}{\delta \theta} \nabla\theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \\ &- \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{1}{\rho} \frac{\delta F}{\delta \theta} \nabla\theta, \frac{\delta H}{\delta \mathbf{u}} \right\rangle, \end{aligned} \tag{4.50}$$

for absolute vorticity defined weakly by

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega} \rangle = \langle \nabla \times \boldsymbol{\Sigma}, \mathbf{u} \rangle + \langle \boldsymbol{\Sigma}, 2\boldsymbol{\Omega} \rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega. \tag{4.51}$$

$\mathbb{V}_\omega$ denotes the vector valued 3D vorticity finite element space given by $V_h^1(\Omega)$ in the compatible framework (see Definition 3). Note that for the 3D Euler test cases in the numerical results section below, the rotation vector $\boldsymbol{\Omega}$ is set to zero. In this section's discussion, the domain Ω is assumed to be boundary-free and as before, we refer to Appendix D for the case with boundaries.

Analogously to the 2D case, we can derive a vorticity evolution equation from the momentum equation, noting that from the compatible framework, $\nabla \times \boldsymbol{\Sigma} \in \mathbb{V}_u$ for any $\boldsymbol{\Sigma} \in \mathbb{V}_\omega$.

Differentiating (4.51) with respect to time, we arrive at

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle = \langle \nabla \times \boldsymbol{\Sigma}, \mathbf{u}_t \rangle \tag{4.52}$$

$$= -\left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot (\nabla \times \boldsymbol{\Sigma}) \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla \times \boldsymbol{\Sigma} \right\rangle \tag{4.53}$$

$$= -\left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla \times \boldsymbol{\Sigma} \right\rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega, \tag{4.54}$$

where similarly to the 2D case, $\nabla \cdot (\nabla \times \boldsymbol{\Sigma}) = 0$. Setting $\boldsymbol{\Sigma} = \boldsymbol{\omega}$, the rate of change of total vorticity evaluates to

$$\frac{1}{2}\frac{d}{dt}\|\boldsymbol{\omega}\|_2^2 = -\left\langle \nabla \times \boldsymbol{\omega}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla \times \boldsymbol{\omega} \right\rangle. \tag{4.55}$$

To incorporate an SUPG stabilisation method, we proceed as before and aim for a residual based formulation. The residual can be derived in an analogous fashion to the thermal case, leading to

$$\boldsymbol{\omega}_{res} = \boldsymbol{\omega}_t + \nabla \times (\boldsymbol{\omega} \times \mathbf{F}/\rho) - \nabla \times (T/\rho \nabla\theta), \tag{4.56}$$

where again we consider projections $\mathbf{F}$ and T in view of the almost Poisson bracket. However, unlike the 2D case, in the 3D residual formulation

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle + \left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \nabla \times \boldsymbol{\Sigma} \right\rangle = \langle \mathcal{P}, \boldsymbol{\omega}_{res} \rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega, \tag{4.57}$$

the choice of upwind stabilisation term $\mathcal{P}$ is less clear, given the more complex vorticity advection term $\nabla \times (\boldsymbol{\omega} \times \mathbf{F}/\rho)$ and the target of introducing dissipation in (4.55). As in the 2D case, for consistency, the resulting term also needs to be incorporated into the momentum equation, which suggests a term of the form

$$\mathcal{P} \coloneqq -\tau(\nabla \times \boldsymbol{\Sigma}) \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}. \tag{4.58}$$

Proposition 31. *Consider a discretised almost Poisson bracket defined by*

$$\begin{aligned}\{F, H\} = &-\left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega}^* \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle \\ &- \left\langle \frac{\delta F}{\delta \rho}, \nabla \cdot \frac{\delta H}{\delta \mathbf{u}} \right\rangle - \left\langle \frac{1}{\rho}\frac{\delta F}{\delta \theta}\nabla\theta, \frac{\delta H}{\delta \mathbf{u}} \right\rangle,\end{aligned} \tag{4.59}$$

for

$$\boldsymbol{\omega}^* = \boldsymbol{\omega} - \tau\boldsymbol{\omega}_{res}, \qquad \boldsymbol{\omega}_{res} = \boldsymbol{\omega}_t + \nabla\times(\boldsymbol{\omega}\times\mathbf{F}/\rho) - \nabla\times(T/\rho\nabla\theta)\,, \tag{4.60}$$

where $\mathbf{F}$ *and* T *are set equal to the projections resulting from the Hamiltonian variations in* $\mathbf{u}$ *and* θ*, respectively, that is*

$$\mathbf{F} = P_{\mathbb{V}_u}(\rho\mathbf{u}), \qquad T = \begin{cases} P_{\mathbb{V}_\theta}\big(\rho(\frac{\rho}{2}+b)\big) & \textit{thermal shallow water equations,} \\ P_{\mathbb{V}_\theta}(c_p\rho\pi) & \textit{Euler equations.} \end{cases} \tag{4.61}$$

Then the momentum equation derived from the corresponding Poisson system is given by

$$\langle\mathbf{w},\mathbf{u}_t\rangle = -\left\langle\mathbf{w},\boldsymbol{\omega}^*\times\frac{1}{\rho}\frac{\delta H}{\delta\mathbf{u}}\right\rangle + \left\langle\frac{\delta H}{\delta\rho},\nabla\cdot\mathbf{w}\right\rangle + \left\langle\frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta,\mathbf{w}\right\rangle \qquad \forall\mathbf{w}\in\mathbb{V}_u, \tag{4.62}$$

which admits a vorticity evolution equation of the form

$$\langle\boldsymbol{\Sigma},\boldsymbol{\omega}_t\rangle + \left\langle\nabla\times\boldsymbol{\Sigma},\boldsymbol{\omega}\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}}\right\rangle - \left\langle\tfrac{1}{\rho}\tfrac{\delta H}{\delta\theta}\nabla\theta,\nabla\times\boldsymbol{\Sigma}\right\rangle = -\left\langle\tau(\nabla\times\boldsymbol{\Sigma})\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}},\boldsymbol{\omega}_{res}\right\rangle \ \ \forall\boldsymbol{\Sigma}\in\mathbb{V}_\omega. \tag{4.63}$$

Proof. *The momentum equation follows directly from the Poisson system. To obtain the vorticity evolution equation, we follow the derivation of the weak form* (4.54)*, and arrive at*

$$\langle\boldsymbol{\Sigma},\boldsymbol{\omega}_t\rangle = \langle\nabla\times\boldsymbol{\Sigma},\mathbf{u}_t\rangle = -\left\langle\nabla\times\boldsymbol{\Sigma},\boldsymbol{\omega}^*\times\frac{1}{\rho}\frac{\delta H}{\delta\mathbf{u}}\right\rangle + \left\langle\frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta,\nabla\times\boldsymbol{\Sigma}\right\rangle \tag{4.64}$$

$$= -\left\langle\nabla\times\boldsymbol{\Sigma},(\boldsymbol{\omega}-\tau\boldsymbol{\omega}_{res})\times\frac{1}{\rho}\frac{\delta H}{\delta\mathbf{u}}\right\rangle + \left\langle\frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta,\nabla\times\boldsymbol{\Sigma}\right\rangle \tag{4.65}$$

$$= -\left\langle\nabla\times\boldsymbol{\Sigma},\boldsymbol{\omega}\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}}\right\rangle + \left\langle\tfrac{1}{\rho}\tfrac{\delta H}{\delta\theta}\nabla\theta,\nabla\times\boldsymbol{\Sigma}\right\rangle - \left\langle\tau(\nabla\times\boldsymbol{\Sigma})\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}},\boldsymbol{\omega}_{res}\right\rangle, \tag{4.66}$$

for all functions $\boldsymbol{\Sigma}\in\mathbb{V}_\omega$*. Moving the advection and baroclinic forcing terms (first two terms on the right-hand side in* (4.66)*) to the left-hand side then yields the required form.* □

Note that while this method is analogous to the 2D setup, it is not clear if it can strictly speaking still be considered as an SUPG method, in the sense that we have not analysed the resulting diffusive effect on the total amount of absolute vorticity. For the modified bracket defined in the proposition above, its rate of change is given by

$$\tfrac{1}{2}\tfrac{d}{dt}\|\boldsymbol{\omega}\|_2^2 = -\left\langle\nabla\times\boldsymbol{\omega},\boldsymbol{\omega}\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}}\right\rangle + \left\langle\tfrac{1}{\rho}\tfrac{\delta H}{\delta\theta}\nabla\theta,\nabla\times\boldsymbol{\omega}\right\rangle - \left\langle\tau(\nabla\times\boldsymbol{\omega})\times\tfrac{1}{\rho}\tfrac{\delta H}{\delta\mathbf{u}},\boldsymbol{\omega}_{res}\right\rangle. \tag{4.67}$$

In this thesis, we omit a rigorous analysis of the last term's diffusive effect, and substantiate the resulting scheme's stabilising feature purely by means of numerical studies, as presented in Section 4.3 below.

Remark 32. *The Poisson bracket framework used in this thesis was originally considered in [MG80] for the ideal compressible magnetohydrodynamic (MHD) equations, which are based on a Hamiltonian of the form*

$$H = \int_\Omega \left(\frac{1}{2}\rho|\mathbf{u}|^2 + \frac{p(\rho, s)}{\gamma - 1} + \frac{1}{2}|\mathbf{B}|^2 \right) dx, \tag{4.68}$$

for velocity $\mathbf{u}$*, density* ρ*, entropy* s*, divergence-free magnetic field* $\mathbf{B}$*, and pressure defined according to*

$$s = c_v ln(\rho^{-\gamma} p) + c_0, \tag{4.69}$$

where c_0 *is a free constant. Further,* c_v *and* γ *denote the specific heat at constant volume and the ratio of specific heats at constant volume and pressure, respectively. The bracket is identical to the standard one given in Definition 1.57 (up to the choice of thermal field variable), together with an additional magnetic field related antisymmetric pair of terms*

$$+ \left\langle \nabla \times \frac{\delta H}{\delta \mathbf{B}}, \mathbf{B} \times \frac{1}{\rho} \frac{\delta F}{\delta \mathbf{u}} \right\rangle - \left\langle \nabla \times \frac{\delta F}{\delta \mathbf{B}}, \mathbf{B} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle. \tag{4.70}$$

Like the absolute vorticity, the magnetic field can be placed in the curl space $V_h^1(\Omega)$ *in the compatible finite element setup (see Definition 3). Setting* $F = \langle \mathbf{\Sigma}, \mathbf{B} \rangle$ *in the Poisson system* (1.53) *then leads to a magnetic field evolution equation*

$$\langle \mathbf{\Sigma}, \mathbf{B}_t \rangle = - \left\langle \nabla \times \mathbf{\Sigma}, \mathbf{B} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle \qquad \forall \mathbf{\Sigma} \in \mathbb{V}_B, \tag{4.71}$$

for $\mathbb{V}_B = V_h^1(\Omega)$*. Note that given the compatible framework, this discretisation satisfies a weak form of the divergence-free condition. To see this, we define a weak divergence on the continuous space* $V_h^0(\Omega)$ *in the 3D compatible setup as*

$$\langle \eta, \tilde{\nabla} \cdot \mathbf{B} \rangle = -\langle \nabla \eta, \mathbf{B} \rangle \qquad \forall \eta \in V_h^0(\Omega). \tag{4.72}$$

We then find that

$$\langle \eta, (\tilde{\nabla} \cdot \mathbf{B})_t \rangle = -\langle \nabla \eta, \mathbf{B}_t \rangle = \left\langle \nabla \times \nabla \eta, \mathbf{B} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle = 0 \qquad \forall \eta \in V_h^0(\Omega), \tag{4.73}$$

where we used $\nabla \times \nabla \eta = 0$*. In particular, this implies that if* $\tilde{\nabla} \cdot \mathbf{B}$ *is zero initially, it will remain so for all time, up to the time discretisation.*

The new methods presented in this thesis for the density, thermal and vorticity fields can equally be applied in the context of MHD. In particular, while no standard atmospheric test cases with sharp pressure gradients are considered here, such gradients commonly appear in MHD. Further, the residual based vorticity approach presented in this chapter can also be adapted to the magnetic field, according to a modified bracket pair of the form

$$+\left\langle \nabla\times\frac{\delta H}{\delta \mathbf{B}}, \mathbf{B}\times\frac{1}{\rho}\frac{\delta F}{\delta \mathbf{u}}\right\rangle \qquad +\left\langle \nabla\times\frac{\delta H}{\delta \mathbf{B}}, \left(\mathbf{B}-\tau\mathbf{B}_{res}\right)\times\frac{1}{\rho}\frac{\delta F}{\delta \mathbf{u}}\right\rangle \tag{4.74}$$

$$\longrightarrow$$

$$-\left\langle \nabla\times\frac{\delta F}{\delta \mathbf{B}}, \mathbf{B}\times\frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle \qquad -\left\langle \nabla\times\frac{\delta F}{\delta \mathbf{B}}, \left(\mathbf{B}-\tau\mathbf{B}_{res}\right)\times\frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle, \tag{4.75}$$

for $\mathbf{B}_{res}=\mathbf{B}_t+\nabla\times(\mathbf{B}\times\mathbf{F}/\rho)$*. Note that importantly, this modification does not interfere with the derivation of* (4.73)*, i.e. maintains the zero weak divergence property. The energy conserving and stabilising properties of the resulting fully upwinded, almost Poisson bracket based scheme for ideal compressible MHD can be verified e.g. for the Orzag-Tang test case [OT79]. Given this thesis' focus on numerical weather prediction, we omit the corresponding results and discussion here.*

4.2 Time discretisation

Before moving on to presenting numerical results, we note that the above discussion is restricted to the PDEs' discretisation in space. From the point of view of the time discretisation using the energy conserving Poisson integrator (1.116), there are two technical observations to be made, which are not considered explicitly in [BC18] and are therefore elaborated here. Using these observations, we then move on to describe the fully discretised schemes to be used in the numerical results section.

For this purpose, we consider the thermal two-dimensional case given in Proposition 29, which we recall contains an SUPG modified advection term of the form

$$-\left\langle \frac{\delta F}{\delta \mathbf{u}}, \left(q-\frac{\tau}{\rho}q^{\theta}_{res}\right)\frac{\delta H}{\delta \mathbf{u}}^{\perp}\right\rangle, \qquad q^{\theta}_{res}=(q\rho)_t+\nabla\cdot(q\mathbf{F})-\nabla^{\perp}\cdot(T/\rho\nabla\theta)\,. \tag{4.76}$$

First, note that in q^{θ}_{res} the flux is given by $\mathbf{F}=P_{\mathbb{V}_u}(\rho\mathbf{u})$ and not $\frac{\delta H}{\delta \mathbf{u}}$. In particular, while these two terms are equal before applying the time discretisation, when using the Poisson

time integrator they evaluate to

$$\bar{\mathbf{F}} = P_{\mathbb{V}_u}(\bar{\mathbf{u}}\bar{\rho}) \quad \neq \quad \frac{1}{3}P_{\mathbb{V}_u}\big(\rho^n\mathbf{u}^n + \frac{1}{2}\rho^n\mathbf{u}^{n+1} + \frac{1}{2}\rho^{n+1}\mathbf{u}^n + \rho^{n+1}\mathbf{u}^{n+1}\big) = \overline{\frac{\delta H}{\delta \mathbf{u}}}, \tag{4.77}$$

for midpoint time averages $\bar{\rho}$ and $\bar{\mathbf{u}}$, and where we used (1.117) in Section 1.2.4 for the time-discretised variational derivative. This holds since from the point of view of the Poisson integrator, q^θ_{res} is part of the transformation J (see (1.116)), so that $\mathbf{F}$ should be discretised according to a midpoint rule. The same holds true for the projection T, i.e. $\bar{T} \neq \frac{\overline{\delta H}}{\delta \theta}$, where the former is discretised according to a midpoint time average, while the latter is time-discretised according to (1.117).

Second, q^θ_{res} also contains the time derivative $(q\rho)_t$, and as noted in Chapter 3, the transformation J should be free of such derivatives in order for the Poisson integrator to be applicable. To approach this problem, we can use an advection operator and an SUPG type operator based on the vorticity evolution equation, defined in an analogous manner to θ^a in Definition 17 and $\mathbb{S}_{\mathbf{u}}$ in Definition 18. Rewriting the evolution equation (4.36) as

$$\begin{aligned}\left\langle \eta + \tau\frac{\delta H}{\delta \mathbf{u}}/\rho \cdot \nabla\eta, (q\rho)_t \right\rangle = -\Bigg(&\left\langle \eta, \nabla\cdot\left(q\frac{\delta H}{\delta \mathbf{u}}\right)\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \nabla^\perp\eta\right\rangle \\ &+ \left\langle \tau\frac{\delta H}{\delta \mathbf{u}}/\rho\cdot\nabla\eta, \nabla\cdot(q\mathbf{F}) - \nabla^\perp\cdot(T/\rho\nabla\theta)\right\rangle\Bigg),\end{aligned} \tag{4.78}$$

for all $\eta \in \mathbb{V}_q$, we find that these operators should be set according to

$$\begin{aligned}\left\langle \eta, q^a\left(\frac{\delta H}{\delta \mathbf{u}}, \frac{\delta H}{\delta\theta}\right)\right\rangle = -\Bigg(&\left\langle \eta, \nabla\cdot\left(q\frac{\delta H}{\delta \mathbf{u}}\right)\right\rangle + \left\langle \frac{1}{\rho}\frac{\delta H}{\delta\theta}\nabla\theta, \nabla^\perp\eta\right\rangle \\ &+ \left\langle \tau\frac{\delta H}{\delta \mathbf{u}}/\rho\cdot\nabla\eta, \nabla\cdot(q\mathbf{F}) - \nabla^\perp\cdot(T/\rho\nabla\theta)\right\rangle\Bigg) \quad \forall \eta \in \mathbb{V}_q,\end{aligned} \tag{4.79}$$

and

$$\left\langle \eta, \mathbb{S}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\big((q\rho)_t\big)\right\rangle = \left\langle \eta + \tau\frac{\delta H}{\delta \mathbf{u}}/\rho\cdot\nabla\eta, (q\rho)_t\right\rangle \qquad \forall \eta \in \mathbb{V}_q, \tag{4.80}$$

noting that like θ^a, the operator q^a implicitly also depends on the state $\mathbf{z} = (\mathbf{u}, \rho, \theta)$. Applying these operators to (4.78) then yields a point-wise version of the form

$$(q\rho)_t = \mathbb{S}^{-1}_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\left(q^a\left(\frac{\delta H}{\delta \mathbf{u}}, \frac{\delta H}{\delta\theta}\right)\right), \tag{4.81}$$

which is analogous to the discussion at (3.13) in the previous chapter.

However, considering the advection term (4.76), we find that $(q\rho)_t$ is a factor of $\frac{\delta H}{\delta \mathbf{u}}^{\perp}$. Similarly to the argument made in Remark 27, we find that if we replace the time derivative by (4.81), we would obtain a non-bilinear bracket. This is also not compatible with the Poisson integrator, and as a workaround, rather than using (4.81), we instead replace $(q\rho)_t$ in q^{θ}_{res} by

$$(q\rho)_t = \mathbb{S}^{-1}_{\mathbf{F}/\rho}\left(q^a\left(\mathbf{F}, T\right)\right), \tag{4.82}$$

where $\mathbf{F}$ and T (defined according to (4.34)) are part of the bracket and not derived from the Hamiltonian. Note that this equality still holds since before time discretisation, $\mathbf{F}$ and $\frac{\delta H}{\delta \mathbf{u}}$ as well as T and $\frac{\delta H}{\delta \theta}$ are equal. In other words, to avoid the occurrence of time derivatives in the Poisson bracket, we will use a vorticity residual of the form

$$q^{\theta}_{res} = \mathbb{S}^{-1}_{\mathbf{F}/\rho}\left(q^a\left(\mathbf{F}, T\right)\right) + \nabla\cdot\left(q\mathbf{F}\right) - \nabla^{\perp}\cdot\left(T/\rho\nabla\theta\right). \tag{4.83}$$

Finally, we time discretise the corresponding almost Poisson bracket by reverting to an approximately energy conserving Poisson integrator, where we specify the time-discretised version of (4.83) according to

$$\bar{q}^{\theta}_{res} \coloneqq \mathbb{S}^{-1}_{\bar{\mathbf{F}}'/\bar{\rho}}\left(q^a\left(\bar{\mathbf{F}}', \bar{T}'\right)\right) + \nabla\cdot\left(\bar{q}\bar{\mathbf{F}}\right) - \nabla^{\perp}\cdot\left(\bar{T}/\bar{\rho}\nabla\bar{\theta}\right), \tag{4.84}$$

for $\bar{\mathbf{F}} = P_{\mathbb{V}_u}(\bar{\rho}\bar{\mathbf{u}})$ and midpoint averages $\bar{\rho}$, $\bar{\mathbf{u}}$, and $\bar{q}$. Similarly, $\bar{T}$ is set according to its definition (4.34), with ρ and θ also time-discretised as midpoint averages. Further, the time-discretised flux $\bar{\mathbf{F}}'$ is specified as

$$\bar{\mathbf{F}}' = \frac{1}{3}P_{\mathbb{V}_u}\big(\rho^n\mathbf{u}^n + \frac{1}{2}\rho^n\mathbf{u}^{n+1} + \frac{1}{2}\rho^{n+1}\mathbf{u}^n + \rho^{n+1}\mathbf{u}^{n+1}\big) = \overline{\frac{\delta H}{\delta \mathbf{u}}}, \tag{4.85}$$

and similarly, $\bar{T}'$ is also time-discretised such that it is equal to the time-discretised Hamiltonian variation $\overline{\frac{\delta H}{\delta \theta}}$. Note that to arrive at a fully energy conserving setup, we would instead have to use the midpoint average $\bar{\mathbf{F}}$ for all flux terms in $\bar{q}^{\theta}_{res}$ – and similarly $\bar{T}$ instead of $\bar{T}'$ – since $\bar{q}^{\theta}_{res}$ is part of the Jacobian J. The choice of $\bar{\mathbf{F}}'$ and $\bar{T}'$ in the first term of $\bar{q}^{\theta}_{res}$ constitutes a deviation from the fully energy conserving Poisson integrator, which crucially enables us to make a substitution of the form

$$\mathbb{S}^{-1}_{\bar{\mathbf{F}}'/\bar{\rho}}\left(q^a\left(\bar{\mathbf{F}}'\right)\right) = \frac{q^{n+1}\rho^{n+1} - q^n\rho^n}{\Delta t}. \tag{4.86}$$

To see this, we can apply an argument analogous to the proof of Proposition 20 in Chapter 3. Note that by another argument analogous to the one described in the paragraph immediately below (3.46), this substitution does not work if we use the midpoint averages $(\bar{\mathbf{F}}, \bar{T})$ instead of $(\bar{\mathbf{F}}', \bar{T}')$ in the first term of $\bar{q}^{\theta}_{res}$. Finally, we also note that we assume a discrete diagnostic vorticity relation of the form

$$\left\langle \eta, \frac{q^{n+1}\rho^{n+1} - q^n\rho^n}{\Delta t} \right\rangle = -\left\langle \nabla^{\perp}\eta, \frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} \right\rangle, \tag{4.87}$$

in order to derive a fully discretised vorticity evolution equation from the fully discretised momentum equation.

The residual term resulting from this substitution, i.e.

$$\bar{q}^{\theta}_{res} = \frac{q^{n+1}\rho^{n+1} - q^n\rho^n}{\Delta t} + \nabla \cdot (\bar{q}\bar{\mathbf{F}}) - \nabla^{\perp} \cdot \left(\bar{T}/\bar{\rho}\nabla\bar{\theta}\right), \tag{4.88}$$

is used implicitly in [BC18] (up to the thermal term), and the above derivation demonstrates that this choice is energy conserving up to the difference between $(\bar{\mathbf{F}}, \bar{T})$ and $(\bar{\mathbf{F}}', \bar{T}')$. Note that this is analogous to the difference (3.53) in the approximately energy conserving scheme in Chapter 3, in the sense that there we replaced $\bar{\mathbf{F}}' = \overline{\frac{\delta H}{\delta \mathbf{u}}}$ by $\bar{\mathbf{F}} = \overline{\frac{\delta H}{\delta \mathbf{u}}}'$ (see (3.42)) in order to be able to make a substitution of the form (3.50), i.e.

$$\mathbb{S}^{-1}_{\bar{\mathbf{u}}}\left(\theta^a(\bar{\mathbf{u}})\right) = \frac{\theta^{n+1} - \theta^n}{\Delta t}. \tag{4.89}$$

Remark 33. *The substitution* (4.89) *is mentioned here purely to highlight the analogy between the vorticity and thermal field SUPG methods in the time discretisation context. In order to focus on the vorticity SUPG setup introduced in this chapter, we apply the fully energy conserving Poisson integrator to bracket* (4.48) *for all terms other than the velocity/vorticity advection term* (4.76)*. The latter includes a vorticity residual that is time-discretised in an approximately energy conserving way according to* (4.88)*. In particular, the fully energy conserving Poisson integrator is also applied to the bracket's thermal field part (last two lines of* (4.48)*), and following the thermal field discussion at* (3.22) *in Chapter 3, the adjoint auxiliary operators* $(\theta^a)^*$ *and* $(\mathbb{S}^{-1}_{\mathbf{u}})^*$ *are solved for at every Picard iteration.*

Before presenting the resulting scheme used in the numerical results section below, we note that in Chapter 3, the substitution (4.89) *was made in the context of a bracket that contains the velocity recovery operator* $\mathbb{U}$*. As mentioned in Remark 30 above, there is no*

straightforward way to derive a vorticity evolution equation if the latter operator occurs in the almost Poisson bracket. In the remainder of this remark, we explain how to obtain a substitution for the thermal field akin to the above form (4.89) *in the context of a Poisson bracket that does not contain* $\mathbb{U}$*. Again, we note that such a substitution will not be used in the numerical results section, and the derivation below is to demonstrate that the auxiliary operators* $(\theta^a)^*$ *and* $(\mathbb{S}_{\mathbf{u}}^{-1})^*$ *can also be avoided in a context where* $\mathbb{U}$ *is not included in the bracket.*

For this purpose, consider a modification to the thermal field SUPG bracket given in Definition 19, whose thermal field part is redefined according to

$$\begin{aligned}\{F,H\}_\theta = \quad & L^\theta_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\Big(\theta;\Big(\frac{\delta H}{\delta \theta}\Big)_{\mathbf{F}/\rho}\Big)+\Big\langle\Big(\widetilde{\widetilde{\frac{\delta H}{\delta\theta}}}\Big)_{\frac{\delta F}{\delta \mathbf{u}}/\rho},\mathbb{S}^{-1}_{\mathbf{F}/\rho}\Big(\theta^a\big(\mathbf{F}/\rho\big)\Big)\Big\rangle \\ & - L^\theta_{\frac{\delta H}{\delta \mathbf{u}}/\rho}\Big(\theta;\Big(\frac{\delta F}{\delta \theta}\Big)_{\mathbf{F}/\rho}\Big)-\Big\langle\Big(\widetilde{\widetilde{\frac{\delta F}{\delta\theta}}}\Big)_{\frac{\delta H}{\delta \mathbf{u}}/\rho},\mathbb{S}^{-1}_{\mathbf{F}/\rho}\Big(\theta^a\big(\mathbf{F}/\rho\big)\Big)\Big\rangle,\end{aligned} \tag{4.90}$$

i.e. we replaced the upwind velocity by $\mathbf{F}/\rho$*. Further, we followed the approximately energy conserving setup as described at* (3.45) - (3.46) *in Section 3.2.2, and swapped the upwind velocity with* $\frac{\delta F}{\delta \mathbf{u}}/\rho$ *and* $\frac{\delta H}{\delta \mathbf{u}}/\rho$ *in the last terms of the first and second line in* (4.90)*, respectively. If we then time-discretise all occurrences of* $\mathbf{F}$ *in* (4.90) *according to* $\bar{\mathbf{F}}'$*, the proof of Proposition 23 works as before, and we find*

$$\mathbb{S}^{-1}_{\bar{\mathbf{F}}'/\bar{\rho}}\Big(\theta^a\big(\bar{\mathbf{F}}'/\bar{\rho}\big)\Big)=\frac{\theta^{n+1}-\theta^n}{\Delta t}. \tag{4.91}$$

Note that we can combine this choice of thermal field part $\{F,H\}_\theta$ *with the fully discretised vorticity SUPG setup as described above. In this case, we arrive at a fully discretised scheme that is energy conserving up to the difference between* $\bar{\mathbf{F}}$ *and* $\bar{\mathbf{F}}'$ *as well as* $\bar{T}$ *and* $\bar{T}'$*.*

In particular, when compared to a corresponding non-energy conserving scheme such as (3.57) - (3.59) *(together with a midpoint time discretisation), we find that the only additional computational effort for this approximately energy conserving scheme lies in the two projections* $\bar{\mathbf{F}}'$ *and* $\bar{T}'$ *that have to be computed in each Picard iteration. Depending on the choice of density upwinding, another DG projection corresponding to* $\overline{\frac{\delta H}{\delta \rho}}$ *is also required.*

4.2.1 Fully discretised schemes

In the numerical results section below, we will consider a bracket of the form (4.48), together with the vorticity SUPG methods as described in Propositions 29 and 31, as well as the velocity upwind method as given in (2.27) (including its 3D version). The vorticity SUPG based bracket is time-discretised as described above, with a modified Poisson integrator that is *approximately energy conserving*. The approximation lies in the choice of discretised residual $\bar{q}^{\theta}_{res}$, which is energy conserving up to the difference between $\bar{\mathbf{F}}$ and $\bar{\mathbf{F}}'$ as well as $\bar{T}$ and $\bar{T}'$. Note that next to this approximation, the resulting scheme contains further approximations to energy conservation given by additional quadrature and solver related limitations as detailed in Section 1.2.4. The scheme is given by

$$\Big\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n \Big\rangle + \Delta t \Bigg(\left\langle \mathbf{w}, \bar{q}^* \overline{\frac{\delta H}{\delta \mathbf{u}}}^{\perp} \right\rangle - \left\langle \overline{\frac{\delta H}{\delta \rho}}, \nabla \cdot \mathbf{w} \right\rangle$$

$$- L^{\theta}_{\mathbf{w}/\bar{\rho}} \Big(\bar{\theta}; \Big(\overline{\frac{\delta H}{\delta \theta}} \Big)_{\bar{\mathbf{u}}} \Big) - \left\langle \mathbf{s}\Big(\overline{\overline{\frac{\delta H}{\delta \theta}}} \Big), \mathbf{w} \right\rangle \Bigg) = 0 \quad \forall \mathbf{w} \in \mathbb{V}_u, \tag{4.92}$$

$$\left\langle \eta, \frac{q^{n+1}\rho^{n+1} - q^n \rho^n}{\Delta t} + \nabla \cdot \left(\bar{q} \overline{\frac{\delta H}{\delta \mathbf{u}}} \right) \right\rangle$$

$$+ \left(L^{\theta}_{\nabla^{\perp}\eta/\bar{\rho}} \Big(\bar{\theta}; \Big(\overline{\frac{\delta H}{\delta \theta}} \Big)_{\bar{\mathbf{u}}} \Big) + \left\langle \mathbf{s}\Big(\overline{\overline{\frac{\delta H}{\delta \theta}}} \Big), \nabla^{\perp} \eta \right\rangle \right) = - \left\langle \tau \overline{\frac{\delta H}{\delta \mathbf{u}}} / \bar{\rho} \cdot \nabla \eta, \bar{q}^{\theta}_{res} \right\rangle \qquad \forall \eta \in \mathbb{V}_q, \tag{4.93}$$

$$\Big\langle \phi, \rho^{n+1} - \rho^n \Big\rangle + \Delta t \left\langle \phi, \nabla \cdot \overline{\frac{\delta H}{\delta \mathbf{u}}} \right\rangle = 0 \qquad \forall \phi \in \mathbb{V}_{\rho}, \tag{4.94}$$

$$\Big\langle \gamma_{\bar{\mathbf{u}}}, \theta^{n+1} - \theta^n \Big\rangle + \Delta t L^{\theta}_{\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}} (\bar{\theta}; \gamma_{\bar{\mathbf{u}}}) = 0 \qquad \forall \gamma \in \mathbb{V}_{\theta}, \tag{4.95}$$

where bars for prognostic variables denote midpoint averages, while bars for Hamiltonian variations are defined as in (1.117). The notation for the forcing terms in the velocity equation's last line as well as the advection operator in (4.95) is given as in the previous chapter. Further, $\bar{q}^* = \bar{q} - \tau/\rho \bar{q}^{\theta}_{res}$, for $\bar{q}^{\theta}_{res}$ as given in (4.88). The 3D version follows analogously, with velocity and vorticity advection terms given as described in Proposition 31.

Note that the resulting system of equations can be solved using a Picard iteration method

as described in the previous chapters (see Section 2.2.1), up to the vorticity field and the advected velocity contribution. For completeness, we illustrate the setup for the latter two parts here. For ease of notation, we abbreviate the thermal forcing terms in (4.93) as $G^\theta(\nabla^\perp\eta)$. First, we solve for the advected vorticity $q^a \in \mathbb{V}_q$ – not to be confused with the vorticity advection operator used at (4.79) above – according to

$$\left\langle \eta, \frac{q^a\rho^{n+1,k} - q^n\rho^n}{\Delta t} + \nabla\cdot(\hat{q}\overline{\frac{\delta H}{\delta \mathbf{u}}}) \right\rangle + G^\theta(\nabla^\perp\eta) + \left\langle \tau\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}\cdot\nabla\eta, \hat{q}^\theta_{res} \right\rangle = 0 \quad \forall \eta \in \mathbb{V}_q, \tag{4.96}$$

where

$$\hat{q} = (q^a + q^n)/2, \qquad \hat{q}^\theta_{res} = \frac{q^a\rho^{n+1} - q^n\rho^n}{\Delta t} + \nabla\cdot(\hat{q}\bar{\mathbf{F}}) - \nabla^\perp\cdot\left(\bar{T}/\bar{\rho}\nabla\bar{\theta}\right), \tag{4.97}$$

and all occasions of the time level $n+1$ in G^θ are set to the known values $n+1, k$. Given q^a, we then solve for the advected velocity $\mathbf{u}^a$ according to

$$\left\langle \mathbf{w}, \frac{\mathbf{u}^a - \mathbf{u}^n}{\Delta t} \right\rangle + \left\langle \mathbf{w}, \left(\hat{q} - \frac{\tau}{\bar{\rho}}\hat{q}_{res}\right)\overline{\frac{\delta H}{\delta \mathbf{u}}}^\perp \right\rangle = 0 \qquad \forall \mathbf{w} \in \mathbb{V}_u. \tag{4.98}$$

The second bracket to be used below, i.e. the one based on velocity upwinding, is coupled with the full Poisson integrator and is therefore *fully energy conserving*, up to quadrature and solver related limitations as mentioned above. The resulting scheme is given by

$$\begin{aligned}\left\langle \mathbf{w}, \mathbf{u}^{n+1} - \mathbf{u}^n \right\rangle + \Delta t \Bigg(&- \left\langle \nabla^\perp\left(\mathbf{w}\cdot\left(\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}\right)^\perp\right), \bar{\mathbf{u}} \right\rangle + \int_\Gamma \left[\left[\mathbf{w}\cdot\left(\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}\right)^\perp \right]\right]\mathbf{n}^\perp\cdot\tilde{\mathbf{u}}\, dS \\ &+ \left\langle \mathbf{w}, f\left(\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}\right)^\perp \right\rangle - \left\langle \overline{\frac{\delta H}{\delta \rho}}, \nabla\cdot\mathbf{w} \right\rangle \\ &- L^\theta_{\mathbf{w}/\bar{\rho}}\left(\bar{\theta}; \left(\overline{\frac{\delta H}{\delta \theta}}\right)_{\bar{\mathbf{u}}}\right) - \left\langle \mathbf{s}\left(\overline{\overline{\frac{\delta H}{\delta \theta}}}\right), \mathbf{w} \right\rangle \Bigg) = 0 \qquad \forall \mathbf{w} \in \mathbb{V}_u, \end{aligned} \tag{4.99}$$

$$\left\langle \phi, \rho^{n+1} - \rho^n \right\rangle + \Delta t \left\langle \phi, \nabla\cdot\overline{\frac{\delta H}{\delta \mathbf{u}}} \right\rangle = 0 \qquad \forall \phi \in \mathbb{V}_\rho, \tag{4.100}$$

$$\left\langle \gamma_{\bar{\mathbf{u}}}, \theta^{n+1} - \theta^n \right\rangle + \Delta t L^\theta_{\overline{\frac{\delta H}{\delta \mathbf{u}}}/\bar{\rho}}(\bar{\theta}; \gamma_{\bar{\mathbf{u}}}) = 0 \qquad \forall \gamma \in \mathbb{V}_\theta, \tag{4.101}$$

and similarly for the 3D case, with perpendicular expressions replaced by the corresponding cross products and curl terms.

In practice, as indicated in the numerical results section below, we find that the approximation to the Poisson integrator by means of (4.88) has no noticeable negative impact on the relative energy error. For this reason, and for a less bulky description of the test cases and results in Section 4.3 below, we will also refer to the scheme (4.92) - (4.95) as energy conserving.

4.3 Numerical results

In this section, we present numerical results for the 2D and 3D vorticity SUPG formulations introduced above. For this purpose, we consider the vorticity SUPG setup (4.92) - (4.95), as well as the velocity upwind setup (4.99) - (4.101). As in the previous two chapters, the test cases were implemented using the automated finite element toolkit Firedrake (see near the bottom of Section 2.2.1). As a small difference to the previous two chapters, we note that in this chapter, the fields depicted in the plots are projections from the underlying compatible spaces into DG and CG spaces which, depending on the underlying compatible space, may be of a higher than first (polynomial) order*.

For the SUPG method applied to the thermal field and 2D as well as 3D vorticity equations, we pick stabilisation parameters of the form

$$\tau_\theta = \left(\lambda_\theta \frac{2}{\Delta t} + \frac{2|\mathbf{u}|}{h}\right)^{-1}, \quad \tau_q = \left(\lambda_q \frac{2}{\Delta t} + \frac{2|\mathbf{u}|}{h}\right)^{-1}, \quad \tau_\omega = \left(\lambda_\omega \frac{2}{\Delta t} + \frac{2|\mathbf{u}|}{h}\right)^{-1}, \tag{4.102}$$

where h corresponds to the cell size and λ_θ, λ_q, and λ_ω are free parameters in the range $[0, 1]$. This choice of parameter can be seen as a compromise between the value $\tau = \Delta t/2$, which corresponds to only a limited amount of diffusion, and the better known value $\tau = h/(2|\mathbf{u}|)$. In our code implementation, we found the latter choice of τ to lead to numerical instability possibly related to round-off errors that negatively affect products of small and large numbers. The free parameter λ can be seen as a weight between these two selections, allowing for more dissipation than $\tau = \Delta t/2$, while avoiding numerical instability. For more details on this topic, see Appendix C.3.1.

In the test cases below, we set $\lambda_\theta = 1/4$, $\lambda_q = 1/3$, and $\lambda_\omega = 1/2$. The only excep-

* This is a consequence of an update in Firedrake; the higher order DG and CG spaces allow for a more accurate representation of the output fields.

tion to this choice of free parameters is given by the higher order 2D rising bubble test case, where the bubble is more well-resolved and the SUPG parameter is therefore set to the less dissipative constant value $\tau_\theta = \Delta t/2$. We note that generally, the resulting fields were found to be somewhat insensitive to the choice of free parameter λ within a range $[1/6, 1/2]$. Values larger than $1/2$ frequently led to an insufficient amount of dissipation, while values closer to 0 often caused oscillations that led to numerical instability in some cases. It should be stressed that the choice of free parameters has not been optimised, and the numerical results should be seen as preliminary with respect to the latter.

4.3.1 Test cases

In what follows, we will consider four test cases. The first describes an advected double vortex scenario for the thermal rotating shallow water equations and is used to confirm the dissipative effect of the vorticity SUPG scheme (4.92) - (4.95). To study its performance when compared to the velocity upwind scheme (4.99) - (4.101), we additionally consider the thermal Williamson 5 test case introduced in Chapter 3. Finally, we move on to the compressible Euler equations, where we compare the vorticity SUPG method to velocity upwinding by means of a 2D and 3D rising bubble scenario akin to the one described in Chapter 3. In all test cases, we will additionally confirm the energy conserving property of the vorticity SUPG method.

2D test cases

The non-dimensionalised thermal advected double vortex test case considered here is based on a non-advected thermal one as described in [EDK19], which in turn is based on a non-thermal one as described in [GHK+09]. The initial conditions are given by velocity, depth, and buoyancy fields of the form

$$\mathbf{u} = \nabla^\perp \psi + \mathbf{u}_0^b, \tag{4.103}$$

$$\rho = \bar{\bar{\rho}} + (f/g)\psi, \tag{4.104}$$

$$\theta = g\big(1 - \epsilon \sin(2\pi y)\big), \tag{4.105}$$

for mean depth, Coriolis parameter, and gravitational acceleration given by $(\bar{\bar{\rho}}, f, g) = (1, 1, 1)$, and buoyancy perturbation magnitude given by $\epsilon = 0.05$. The stream function

and initial background flow are set to

$$\psi = a\left(e^{-\left((x-x_1)^2+(y-y_1)^2\right)/b^2} + e^{-\left((x-x_2)^2+(y-y_2)^2\right)/b^2}\right), \tag{4.106}$$

$$\mathbf{u}_0^b = (0.1, 0), \tag{4.107}$$

where $(a, b) = (0.01, 0.1)$. Further, the domain Ω is given by a periodic unit square. Considering the rotating shallow water equations with initial conditions $(\mathbf{u}, \rho) = (\mathbf{u}_0^b, 0)$, we find that the background flow evolves as $\mathbf{u}^b = 0.1\big(\cos(ft), \sin(ft)\big)$, i.e. in a rotating fashion. In particular, the double vortex arising from the stream function ψ will move in a circle while evolving dynamically due to the vortex-vortex interaction.

To investigate the dissipative property of the vorticity SUPG framework introduced above, we consider the scheme (4.92) - (4.95). The test case is then run twice, with the SUPG parameters set as described above, and again with τ_q set to zero (i.e. no vorticity SUPG). The simulation is run up to $t_{max} = 4\pi$, corresponding to two circular evolutions. The mesh is regular with 64^2 square shaped cells, and we consider the lowest order compatible space triple $(CG_1,\ H(\mathrm{div})_1,\ DG_0)$; see Section 1.1.2. The time step is set to $\Delta t = \pi/500$, and in order to demonstrate energy conservation to machine precision, we perform 16 Picard iterations per time step. Results for the vorticity type field q at time t_{max} are depicted in Figure 4.1, and the corresponding relative energy and enstrophy developments are given in Figure 4.2.

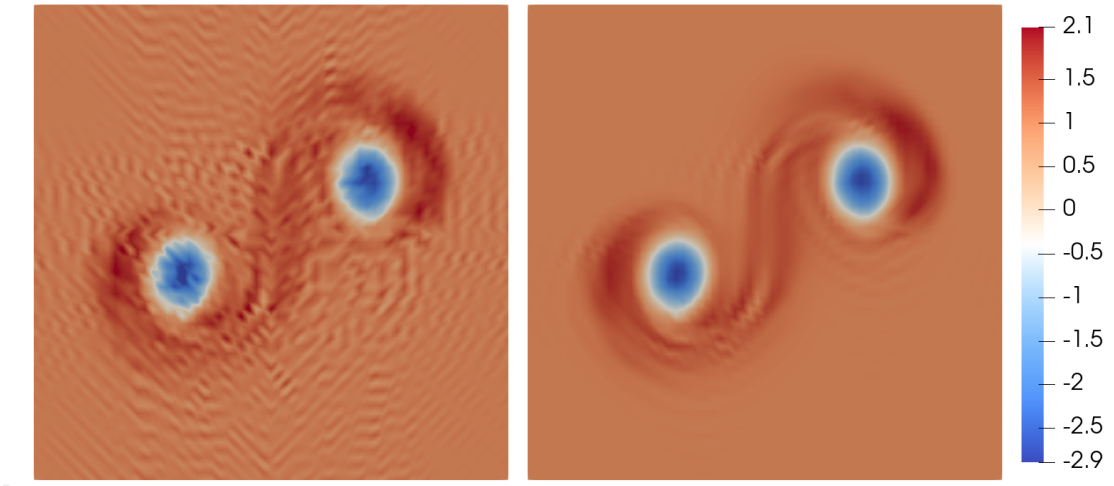


Figure 4.1: Vorticity type field q after two rotations ($t = 4\pi$) for advected thermal double vortex test case. Left: prognostic vorticity setup without SUPG. Right: prognostic vorticity setup with SUPG. 64^2 cells, $\Delta t = \pi/500$, with 16 Picard iterations per time step.

Considering Figure 4.1, we find that the vorticity SUPG method greatly reduces the ex-

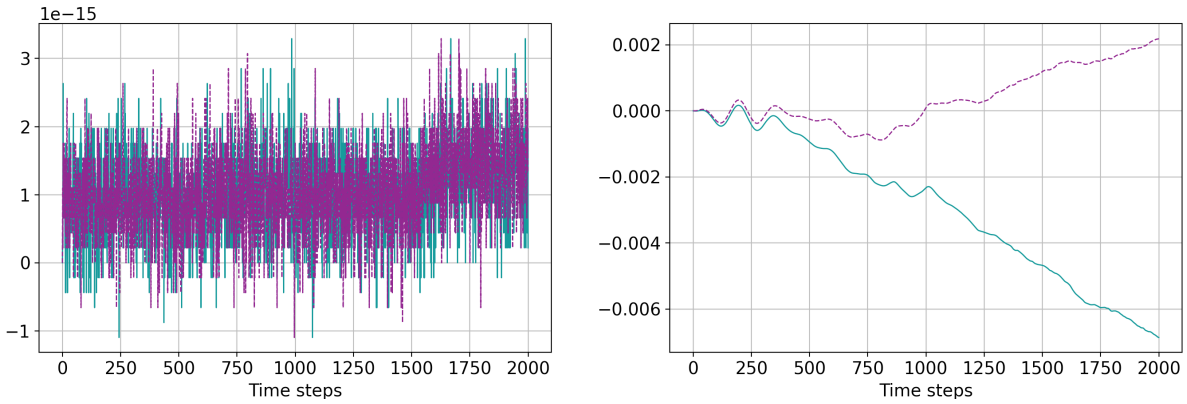


Figure 4.2: Relative energy (left) and enstrophy (right) development for advected thermal double vortex test case. Dashed purple: prognostic vorticity setup without SUPG. Cyan: prognostic vorticity setup with SUPG. 64^2 cells, $\Delta t = \pi/500$, with 16 Picard iterations per time step.

tend and magnitude of oscillations in the vorticity type field q. This is also confirmed by the relative enstrophy evolution depicted in Figure 4.2, where there is a marked decrease in relative enstrophy in the vorticity SUPG run when compared to the one without vorticity SUPG. Finally, as expected, energy is conserved up to machine precision regardless of whether or not vorticity SUPG is included.

Next, we revisit the thermal Williamson 5 test case, as introduced at (3.60) - (3.62) in the previous chapter, which we consider for the purpose of comparing the vorticity SUPG method with velocity upwinding. The mesh is given by an icosahedral triangulation, where as in the previous chapters, refinement level 0 corresponds to 20 triangles. For every higher level, each triangle is refined to 4 triangles (so that each increase corresponds to halving the cell side length Δx). The finite element space triple is set to the lowest order triple $(P_2,\ BDM_1,\ P_0^{DG})$, which we consider for refinement level 5, with $\Delta t = 240$ s and 4 Picard iterations per time step. For additional comparison purposes, we also consider the velocity upwind method together with the next to lowest order triple $(P_3,\ BDM_2,\ P_1^{DG})$, with the same space and time discretisation parameters. The simulation is run for 50 days, and the resulting vorticity type field q is depicted in Figure 4.3. Further, the energy development for the two lowest order runs can be found in Figure 4.5.

Considering the vorticity type field development, we find that the next to lowest order velocity upwind setup and the lowest order vorticity SUPG setup are in very good agreement. Conversely, the lowest order velocity upwind setup preserves fewer fine scale details, contains small scale oscillations throughout the domain, and leads to a somewhat different dynamical development (considering e.g. the large negative vorticity filament in

the bottom right quadrant). Finally, energy is conserved to the order of 10^{-11} for both lowest order schemes, indicating that a relatively smaller number of Picard iterations is sufficient in terms of a good degree of energy conservation for slow, large scale motion.

Moving on to the compressible Euler equations, we consider a rising bubble test case equal to the one as introduced at (3.64) in the previous chapter, with a potential temperature perturbation magnitude given by $\Delta\theta_{mag} = 7.5$ K. As with the previous test case, we consider the vorticity SUPG and velocity upwind setups for a set of lowest order spaces, i.e. $(CG_1,\ H(\mathrm{div})_1,\ DG_0)$. Following the previous chapter, we further consider an extruded mesh. For comparison purposes, we also consider a velocity upwind setup with a higher order set of spaces $(CG_3,\ H(\mathrm{div})_3,\ DG_2)$. In both cases, the compatible space triple is paired with the according Charney-Phillips type potential temperature finite element space. The mesh consists of 160^2 cells, with $\Delta t = 0.5$ s for the lowest order space setup and $\Delta t = 0.2$ s for the higher order one. To verify energy conservation, we perform 16 Picard iterations for the lowest order setup. For the higher order control run, 4 iterations are used and the thermal SUPG parameter is set to $\tau_\theta = \Delta t/2$. The simulation is run up to $t_{\max} = 600$ seconds, and the resulting potential temperature and vorticity type fields are depicted in Figure 4.4. Further, the relative energy error development for the lowest order runs is given in Figure 4.5.

Considering the potential temperature fields, we find that both lowest order setups exhibit a significant amount of diffusion due to the first order classical DG upwind method applied in the horizontal direction, given the Charney-Phillips type temperature space. However, the vorticity SUPG formulation resembles the high order setup better in the sense that the two trailing vortices are at the same position, while for the velocity upwind formulation they are too low. In the vorticity type field q plots, we additionally find that in the case of the vorticity SUPG setup, both main trailing vortices as well as the ones arising in the secondary plume at the bubble's top are free of oscillations, while this is not the case for the velocity upwind setup in the lowest order space context. Further, as expected, both lowest order setups conserve energy up to solver tolerance, with occasional jumps in the relative energy development. This includes the third to last time step for the vorticity SUPG formulation; for the last two time steps the change in relative energy evolves at the order of 10^{-14} again. As noted in the previous chapter, the jumps are probably due to an insufficient number of Picard iterations at the time steps where such jumps occur. Finally, note that all runs exhibit oscillations in the vorticity type field q near the boundaries, including the periodic vertical one. We will find below that these

do not occur in the 3D case – or at least with a much smaller amplitude – and further investigations are required to clarify why they occur here.

3D test case

In the final test case of this thesis, we apply the upwind stabilised compatible finite element framework to a three dimensional scenario. It is given by a 3D version of the rising bubble test case considered above, following [TD16]. The domain is defined by a horizontally periodic box $\Omega = [1000\text{ m}, 1000\text{ m}, 1500\text{ m}]$, and the perturbation is specified as before, up to domain specific parameters. They are given by the bubble's centre, located at $[500\text{ m}, 500\text{ m}, 260\text{ m}]$, and its radius as well as perturbation magnitude, which are given by 250 m and $\Delta\theta_{mag} = 2.5$ K, respectively.

As with the previous two test cases, we will compare the vorticity SUPG and velocity upwind formulations, and verify their energy conserving property. For a comparison from the energy conservation point of view, we also consider the non-energy conserving scheme (3.57) - (3.59) in Chapter 3. In terms of finite element spaces, we use the lowest order 3D triple $(H(\text{curl})_1,\ H(\text{div})_1,\ DG_0)$ (see Figure 1.4 in Chapter 1), and a fully continuous potential temperature space CG_1 in order to avoid the strongly diffusive effect encountered in the Charney-Phillips type discretisation in the previous section's corresponding 2D test case. In simulations not shown here, we found 3D runs for this test case with the latter type of space to perform badly when lowest order spaces are used, regardless of whether vorticity SUPG or velocity upwinding was applied.

The mesh is obtained by extruding a regular 2D base mesh consisting of squares, and its cells are set to a resolution of $\Delta x = \Delta y = \Delta z = 20$ m. The simulation is run up to $t_{max} = 300$ s, for $\Delta t = 0.8$ s and 4 Picard iterations per time step. The resulting potential temperature fields at time t_{max} are depicted in Figure 4.6, which also includes a reference plot from [TD16]. The latter was computed using a fifth order WENO finite volume scheme, on an irregular mesh with a cell size of 30 m. Further, the vorticity field $\boldsymbol{\omega}$ is depicted in Figure 4.7 for the two energy conserving simulations. Finally, the relative energy error development for the three compatible finite element setups is given in Figure 4.8.

We find that the vorticity SUPG setup is in very good agreement with the reference finite volume scheme, in the sense that the trailing vortices, inner vertical column, and bubble

top are approximately at the same position and have the same shape. Conversely, for the two velocity upwind setups, the trailing vortices are at a lower position, the top exhibits a sharper curvature, and the inner vertical column is less pronounced in-between the trailing vortices. Since we consider a CG space for the potential temperature, none of the three finite element simulations exhibit the over-diffusive effect seen in the 2D case, where a Charney-Phillips type space is used. Further, we note that all four simulations contain negative perturbation values based on spurious oscillations, with the strongest ones occurring between the vertical column and trailing vortices in the vorticity SUPG setup. Altogether, when compared to the reference simulation, it can still be concluded that the latter setup outperforms the velocity upwind stabilised one. Finally, we note that in a simulation not shown here, we also considered the vorticity SUPG setup with the vorticity SUPG parameter τ_ω set to zero. The resulting vorticity field development included clearly visible oscillations at time $t = 100$ s, and the linear solvers eventually failed to converge at $t = 194$ s.

Differences between the vorticity SUPG and velocity upwind setups are also apparent in the vorticity field. While the vertical column and vortex structure are clearly visible in the vorticity SUPG setup, this is less the case in the setup corresponding to velocity upwinding, which also contains oscillations at the two vortices' positions. Finally, as seen in Figure 4.8, the energy conserving formulation improves the relative energy error by two orders of magnitude when compared to the non-energy conserving scheme, noting that 4 Picard iterations per time step were used. As explained in the previous chapter, a better degree of energy conservation can be attained for the energy conserving formulations if more iterations are used.

4.4 Summary

In this chapter, we presented an almost Poisson bracket discretisation which includes a novel SUPG method for the vorticity field advection. The bracket was used to derive discretisations for the thermal shallow water and 2D as well as 3D Euler equations, which are energy conserving up to a small approximation in the Poisson integrator based time discretisation. The incorporation of the SUPG method relies on a residual based formulation used to ensure consistency between the velocity and vorticity equations. We demonstrated the derivation of the governing equations given the underlying SUPG modified almost Poisson bracket for the 2D and 3D case. Additionally, in a series of test cases, we verified the stabilising effect of the SUPG method applied to the vorticity equation.

The series included an advected thermal double vortex scenario for the thermal shallow water equations, which demonstrates the dissipative effect of the SUPG method. Further, it included a more geophysically relevant spherical test case, where the vorticity SUPG setup was applied to a lowest order set of finite element spaces. The resulting vorticity field development is of comparable quality to that of a simulation where the velocity upwind method is applied to the next higher set of spaces. Finally, we considered the vorticity SUPG method for a rising bubble test case for the Euler equations in two and three dimensions, demonstrating that in both cases, the vorticity SUPG method outperforms velocity upwinding when applied to a lowest order set of spaces.

In the three-dimensional case, the stabilising effect of the SUPG method incorporated into the vorticity equation was studied by means of numerical test cases only. In future work, we aim to conduct a more rigorous mathematical analysis on the method's dissipative effect on the vorticity field development. Further, the choice of tuning parameter within the SUPG parameters has so far only been investigated to a very small degree, and we also aim to shed more light on this topic in future work.

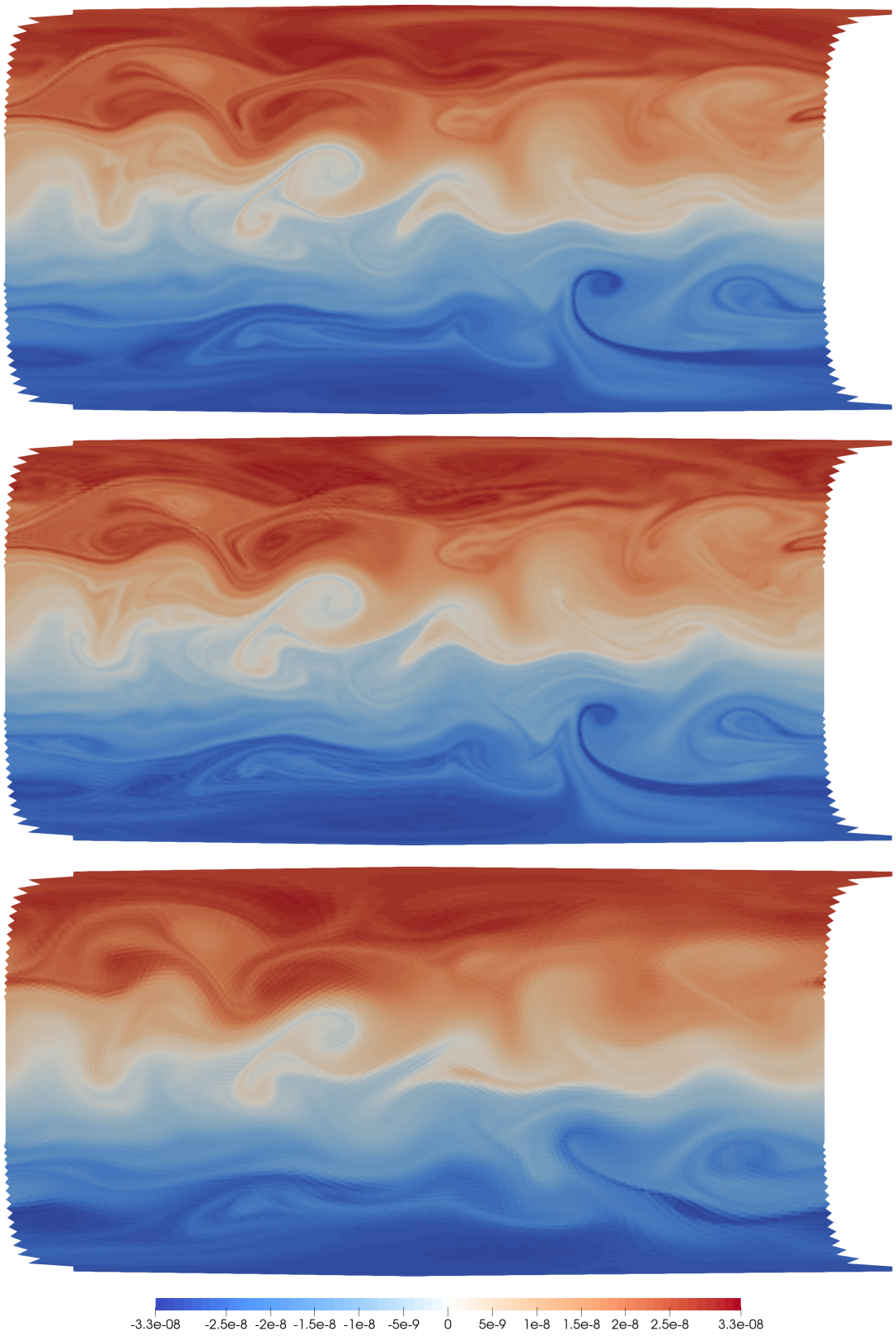


Figure 4.3: Vorticity type field q after 50 days for thermal Williamson 5 test case. Top to bottom: velocity upwind setup with second (polynomial) order velocity function space, vorticity SUPG setup with first order velocity function space, velocity upwind setup with first order velocity function space. Mesh refinement level 5, $\Delta t = 240$ s, with 4 Picard iterations per time step.

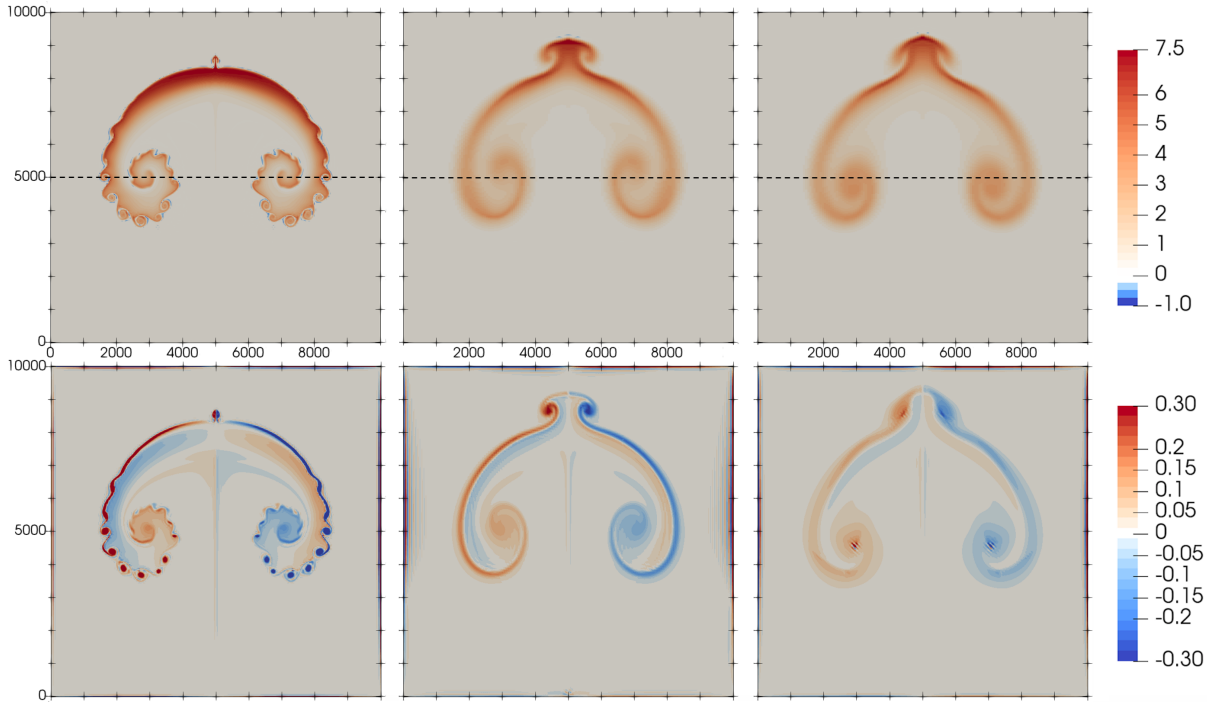


Figure 4.4: Potential temperature fields for 2D rising bubble test case after 600 seconds. Top row: potential temperature. Bottom row: vorticity type field q. Left column: energy conserving setup with velocity upwinding and third (polynomial) order velocity function space. Middle column: energy conserving setup with vorticity SUPG and first order velocity function space. Right column: energy conserving setup with velocity upwinding and first order velocity function space. $\Delta x = \Delta z = 6.25$ m, and $\Delta t = 0.5$ s and 0.2 s for first and third order space setups, respectively, with 16 and 4 Picard iterations per time step for first and third order space setups, respectively. A dashed horizontal line is placed at 5000 metres.

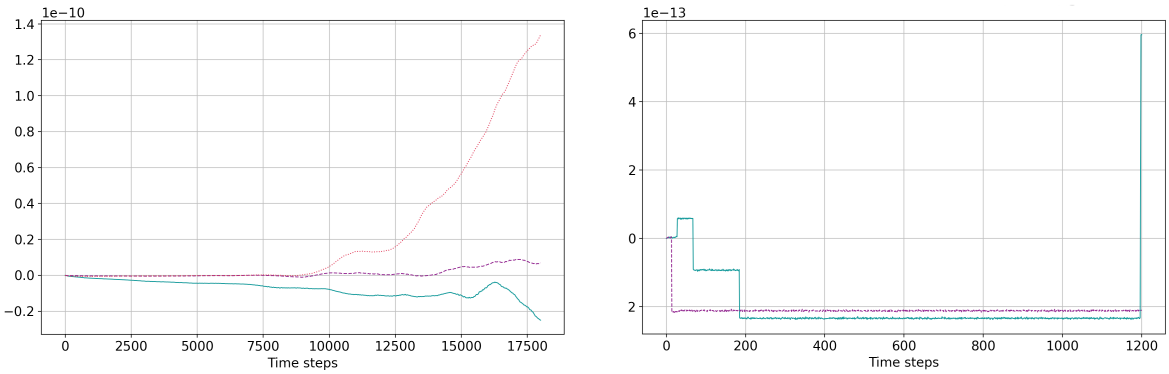


Figure 4.5: Relative energy development for thermal Williamson 5 (left) and 2D rising bubble (right) test cases. Dashed purple: velocity upwind setup. Cyan: vorticity SUPG setup. Dotted red in Williamson 5 plot: higher order velocity setup. 4 and 16 Picard iterations for Williamson 5 and 2D rising bubble test cases, respectively.

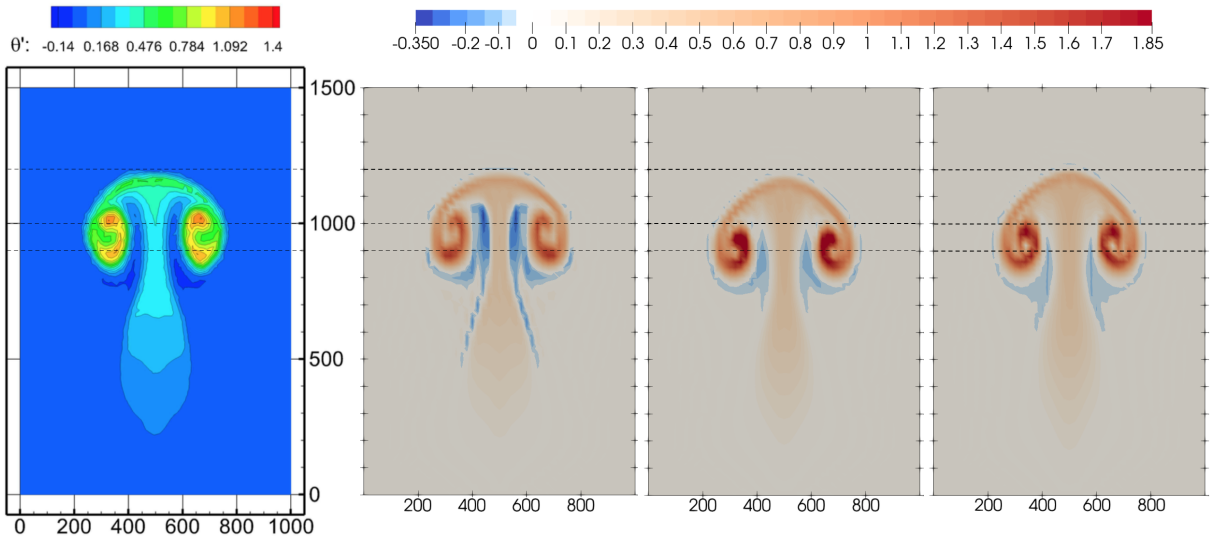


Figure 4.6: Potential temperature fields for 3D rising bubble test case after 300 seconds. Left: reference run from [TD16]. Middle left: energy conserving setup with vorticity SUPG. Middle right: energy conserving setup with velocity upwinding. Right: non-energy conserving setup with velocity upwinding. For the right three images, $\Delta x = \Delta y = \Delta z = 20$ m, $\Delta t = 0.8$ s, with 4 Picard iterations per time step. Dashed horizontal lines are placed at 900, 1000, and 1200 metres.

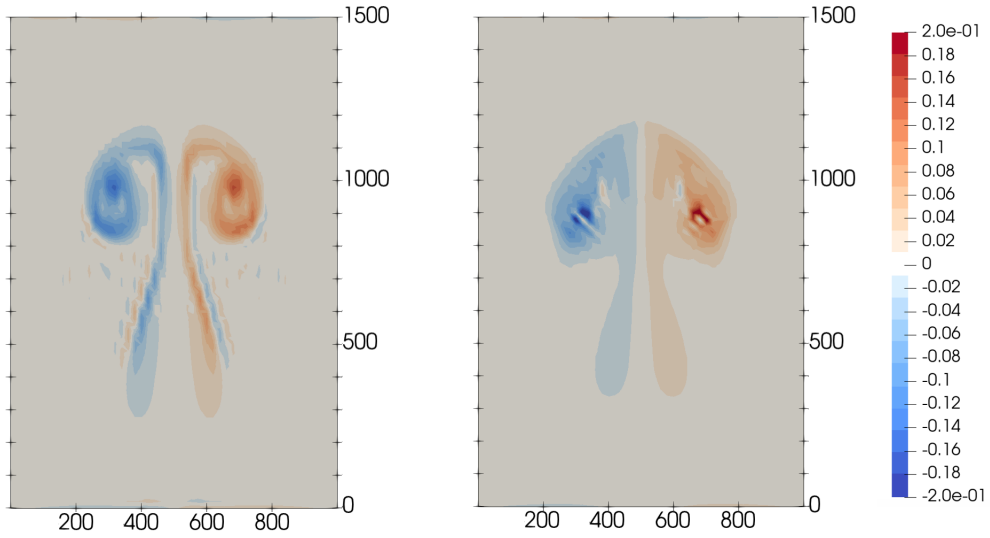


Figure 4.7: Vorticity fields for 3D rising bubble test case after 300 seconds. Left: energy conserving setup with vorticity SUPG. Right: energy conserving setup with velocity upwinding. $\Delta x = \Delta y = \Delta z = 20$ m, $\Delta t = 0.8$ s, with 4 Picard iterations per time step.

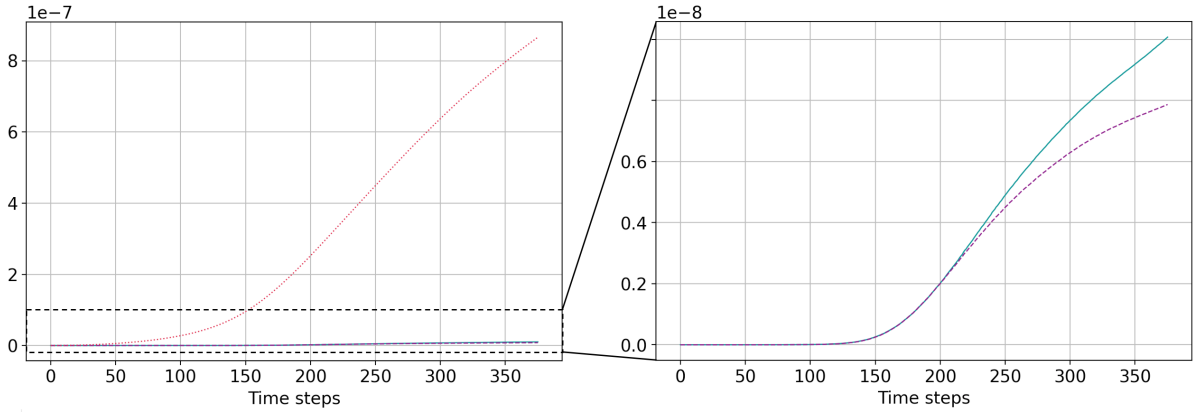


Figure 4.8: Relative energy development for 3D rising bubble test case. Dashed purple: energy conserving velocity upwind setup. Cyan: energy conserving vorticity SUPG setup. Dotted red: non-energy conserving velocity upwind setup. $\Delta x = \Delta y = \Delta z = 20$ m, $\Delta t = 0.8$ s, with 4 Picard iterations per time step.

5

Conclusion

In this thesis, we considered the compatible finite element method, which is currently of interest in numerical weather prediction as it allows for more general meshes, while retaining favourable properties arising from discrete vector calculus identities. The former is of importance due to the need for quasi-uniform meshes, which avoid parallel computation bottlenecks that arise from grid poles. The latter properties are given by increased stability as well as the possibility to avoid unphysical forcing terms and inconsistent systems of equations. Additionally, we considered the Poisson bracket based Hamiltonian framework, which can be used to construct energy conserving space discretisations. Energy conservation is an important quantity in particular for climate models, and a lack thereof can lead to net energy biases and spurious motions. The Hamiltonian framework can be applied to the compatible finite element method, and the combination has been considered in the literature before.

However, energy conservation is often derived at the expense of a lack of transport stabilisation, which is aimed at reducing unphysical oscillations. If such oscillations are not controlled, they may pollute the solution fields and lead to a qualitatively insufficient field development. The aim of this thesis was therefore to formulate energy conserving, compatible finite element based transport stabilisation methods for fields where such methods have not yet been devised in the literature.

Specifically, new energy conserving stabilised methods were designed for the transport

of density related fields, thermal type fields, and vorticity type fields. For the density related fields, an energy conserving classical Discontinuous Galerkin (DG) upwind method and a variation was designed. The formulation for the classical method is based on an operator used to recover the velocity field from the momentum flux that occurs in the Hamiltonian framework. For the thermal related fields, an energy conserving Streamline Upwind Petrov Galerkin (SUPG) method was devised, as well as a variation for partially continuous finite element spaces. This is achieved by means of a stabilisation type and a transport type operator aimed at reformulating the stabilised thermal field transport equation in a suitable way for the Hamiltonian framework. Finally, for the vorticity type fields, an energy conserving SUPG method was derived that is consistent with the momentum equation, from which the vorticity equation is derived. The corresponding approach is based on a residual form of the stabilisation method, allowing the latter to be incorporated into the momentum equation.

Due to the generality of the Hamiltonian framework, these methods are applicable to a range of equation sets relevant to geophysical fluid dynamics and beyond. In particular, in this thesis, we considered the rotating shallow water equations, thermal rotating shallow water equations, and 2D as well as 3D dry ideal compressible Euler equations. For all of the aforementioned new methods, we verified an improved field development when transport stabilisation is included, and further that energy is conserved to machine precision or solver tolerance when the space discretisation is also coupled with an energy conserving time discretisation. This was achieved by means of more basic planar test cases, as well as standard spherical (thermal) rotating shallow water scenarios and Euler scenarios for 2D vertical slices and 3D cubic sections of the atmosphere.

In view of the methods' applicability, we also included discussions and numerical results on approximations to the energy conserving time discretisation. These approximations lead to simplifications in which none of the auxiliary operators introduced in the energy conserving space discretisations have to be computed explicitly. The resulting schemes only require additional projections for the Hamiltonian variations when compared to similar non-energy conserving schemes, leading to only a small increase in additional computational cost. In a setup using the small number of quasi-Newton iterations usually considered in numerical weather prediction, we demonstrated that these approximately energy conserving schemes still lead to an order of magnitude reduction in relative energy error when compared to the corresponding non-energy conserving schemes, and an equally good qualitative field development. In one scenario, given by a falling cold air bubble, we

showed that this increase in accuracy may lead to additional small scale features in the field development, which have also been observed in the literature in a different, sub-grid model based context.

We also compared two transport stabilisation methods within the energy conserving Hamiltonian framework, given by the SUPG method applied to vorticity transport, as well as Lie derivative type upwinding applied to velocity transport. The former method was newly introduced here, while the latter was considered in the literature before. For this purpose, we considered a spherical test case associated with flow past a mountain, as well as 2D and 3D Euler test cases corresponding to a rising hot air bubble. In the case of lowest order compatible finite element sets of spaces, we demonstrated a more accurate field development for the SUPG method applied to vorticity transport. The vorticity field evolution in the spherical test case matched well with the velocity upwind setup applied to the next higher order set of spaces, as did the potential temperature evolution in the 3D Euler test case when compared to a reference finite volume scheme.

Next to providing a path to energy conservation, the Poisson bracket based Hamiltonian framework also reveals the energy transfers occurring between different types of energy. These transfers play an important role in atmospheric dynamics, and unbalanced or unphysical such transfers may lead to the aforementioned energy biases and spurious motions. While left largely unexplored in this thesis, in future work, we aim to study the transfers' representation after discretisation. For instance, this may shed light on the underlying mechanism leading to the formation of small scale features in the cold air bubble scenario when an approximately energy conserving scheme is applied. Further, a better understanding of transfers may facilitate the design of approximately energy conserving schemes aimed at computational efficiency while keeping the resulting energy error minimal.

Finally, there is also a need to develop more efficient nonlinear solvers for Poisson integrators. In this thesis, we solved the nonlinear systems of equations arising from the latter using a Picard iteration method, which relies on the linearisation around a state of rest. In the resulting linearised system, the thermal field was eliminated, and the subsystem in velocity and the density related field was solved for using a hybridised method. Ideally we would use the full Jacobian within Newton's method for faster convergence, but this relies upon the development of efficient preconditioners for the Jacobian – which is not hybridisable, unlike the linearisation around the state of rest.

A

Finite element cell mappings

In practice, for efficiency reasons computations are usually performed on the reference cell $\hat{K}$, using mappings to each cell K of the tessellation, so that only one set of basis functions and quadrature points is needed. Such mappings can be defined e.g. such that normal and tangential components are preserved, allowing for the construction of div and curl conforming spaces. Here, we briefly outline the relevant mappings for the compatible finite element spaces introduced in Section 1.1. Further, we include a brief discussion on a modification to these mappings, called rehabilitation.

Given a mapping $\Phi\colon \hat{K} \to K$ with $\Phi(\hat{\mathbf{x}}) = \mathbf{x}$, and a vector function $\hat{\mathbf{v}}(\hat{\mathbf{x}})$ defined on $\hat{K}$, the contravariant Piola transformation of $\hat{\mathbf{v}}$ is defined by

$$\mathbf{v}(\mathbf{x}) = \frac{\mathbf{J}\hat{\mathbf{v}}(\hat{\mathbf{x}})}{\det \mathbf{J}}, \tag{A.1}$$

for Jacobian transformation $\mathbf{J} = \partial\Phi_i/\partial\hat{x}_j$. Further, the covariant Piola transformation is given by

$$\mathbf{v}(\mathbf{x}) = \mathbf{J}^{-T}\hat{\mathbf{v}}(\hat{\mathbf{x}}), \tag{A.2}$$

for $\mathbf{J}^{-T} = (\mathbf{J}^{-1})^T$. To illustrate the preservation properties of the co- and contravariant mappings, we follow [RKL10] and assume for simplicity Φ to be an affine mapping of the form $\Phi(\hat{\mathbf{x}}) = \mathbf{J}\hat{\mathbf{x}} + \mathbf{b}$, with a triangular reference element $\hat{K}$. For tensor product elements and curvilinear ones used e.g. for spherical domains, we refer to [NSC16]. Consider an edge $\hat{\mathbf{e}}$ of the reference cell $\hat{K}$, which is mapped to an edge $\mathbf{e} = \Phi(\hat{\mathbf{e}})$ of K. The unit

tangents along $\mathbf{e}$ and $\hat{\mathbf{e}}$, respectively, are then given by

$$\mathbf{t} = \mathbf{e}/\|\mathbf{e}\|, \qquad \hat{\mathbf{t}} = \hat{\mathbf{e}}/\|\hat{\mathbf{e}}\|, \tag{A.3}$$

and using the endpoints $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{e}$, we find

$$\|\mathbf{e}\|\mathbf{t} = \mathbf{e} = \mathbf{x}_2 - \mathbf{x}_1 = \mathbf{J}(\hat{\mathbf{x}}_2 - \hat{\mathbf{x}}_1) = \mathbf{J}\hat{\mathbf{e}} = \|\hat{\mathbf{e}}\|\mathbf{J}\hat{\mathbf{t}}. \tag{A.4}$$

For $\mathbf{v}$ mapped via the covariant transformation, this leads to

$$\|\mathbf{e}\|\mathbf{t} \cdot \mathbf{v} = \|\hat{\mathbf{e}}\|\mathbf{J}\hat{\mathbf{t}} \cdot \mathbf{J}^{-T}\hat{\mathbf{v}} = \|\hat{\mathbf{e}}\|\hat{\mathbf{t}} \cdot \hat{\mathbf{v}}, \tag{A.5}$$

and we find that in this case tangential components for vector fields are preserved up to edge lengths. Additionally, using a clockwise rotation matrix $\mathbf{R}$, we can express the corresponding unit normals as $\mathbf{n} = \mathbf{R}\mathbf{t}$ and $\hat{\mathbf{n}} = \mathbf{R}\hat{\mathbf{t}}$, respectively. Applying (A.4), the normals relate to each other via

$$\|\mathbf{e}\|\mathbf{n} = \|\mathbf{e}\|\mathbf{R}\mathbf{t} = \|\hat{\mathbf{e}}\|\mathbf{R}\mathbf{J}\hat{\mathbf{t}} = \|\hat{\mathbf{e}}\|(\det \mathbf{J})\mathbf{J}^{-T}\mathbf{R}\hat{\mathbf{t}} = \|\hat{\mathbf{e}}\|(\det \mathbf{J})\mathbf{J}^{-T}\hat{\mathbf{n}}, \tag{A.6}$$

where we also used

$$\mathbf{R}\mathbf{J}\mathbf{R}^T = (\det \mathbf{J})\mathbf{J}^{-T}. \tag{A.7}$$

We then find for $\mathbf{v}$ mapped via the contravariant transformation,

$$\|\mathbf{e}\|\mathbf{n} \cdot \mathbf{v} = \|\hat{\mathbf{e}}\|(\det \mathbf{J})\mathbf{J}^{-T}\hat{\mathbf{n}} \cdot \frac{\mathbf{J}\hat{\mathbf{v}}}{\det \mathbf{J}} = \|\hat{\mathbf{e}}\|\hat{\mathbf{n}} \cdot \hat{\mathbf{v}}, \tag{A.8}$$

implying that normal components for vector fields are preserved up to edge lengths. In particular, given two adjacent cells within the domain's tessellation, their common edge will evaluate equally on the reference cell with respect to their normal and tangential components, for the contravariant and covariant Piola transformations, respectively. Using the Piola transformations together with the divergence and curl finite elements defined in Section 1.1 therefore leads to the required divergence and curl conforming finite element spaces.

Next to ensuring conforming spaces, the mappings should also be consistent with compatibility, as given in Definition 3 in Section 1.1 (and similar for the 2D case). In particular,

for a function $\mathbf{u} \in H(\text{div};\ \Omega)$, commutativity requires

$$\nabla \cdot (\pi^2 \mathbf{u}) = \pi^3 (\nabla \cdot \mathbf{u}), \tag{A.9}$$

for $\pi^2 \mathbf{u} \in V_h^2(\Omega)$ and $\pi^3(\nabla \cdot \mathbf{u}) \in V_h^3(\Omega)$, and this should also hold after transforming to the reference cell. For the contravariant mapping (A.1), it can be shown that given a velocity field $\mathbf{v}$, the divergence operator transforms as [Cia13]

$$\nabla \cdot \mathbf{v} = \frac{1}{\det \mathbf{J}} \hat{\nabla} \cdot \hat{\mathbf{v}}, \tag{A.10}$$

for differential operator $\hat{\nabla}$ defined on $\hat{K}$, and $\hat{\mathbf{v}} = \mathbf{v} \circ \Phi$. In particular, for the L^2-conforming space, commutativity then implies a cell mapping of the form

$$\phi(\mathbf{x}) = \frac{\hat{\phi}(\hat{\mathbf{x}})}{\det \mathbf{J}}, \tag{A.11}$$

for $\hat{\phi} = \phi \circ \Phi$. To see this, note that for $\phi = \pi^3(\nabla \cdot \mathbf{u})$, the commutativity relation (A.9) maps onto reference cells as

$$\begin{aligned} \nabla \cdot (\pi^2 \mathbf{u}) = \pi^3 (\nabla \cdot \mathbf{u}) \quad &\rightarrow \quad \frac{1}{\det \mathbf{J}} \hat{\nabla} \cdot \widehat{(\pi^2 \mathbf{u})} = \frac{1}{\det \mathbf{J}} \widehat{\pi^3 (\nabla \cdot \mathbf{u})} \\ &\rightarrow \quad \hat{\nabla} \cdot \widehat{(\pi^2 \mathbf{u})} = \widehat{\pi^3 (\nabla \cdot \mathbf{u})}, \end{aligned} \tag{A.12}$$

as required, where we used (A.10) for $\mathbf{v} = \pi^2 \mathbf{u}$. Similarly, for a function $\psi \in H^1(\Omega)$, commutativity requires

$$\nabla(\pi^0 \psi) = \pi^1 (\nabla \psi), \tag{A.13}$$

for $\pi^0 \psi \in V_h^0(\Omega)$ and $\pi^1(\nabla \psi) \in V_h^1(\Omega)$. In this case, given a scalar function χ, the correct mapping to ensure commutativity after transforming to the reference cell is given by

$$\chi(\mathbf{x}) = \hat{\chi}(\hat{\mathbf{x}}). \tag{A.14}$$

for $\hat{\chi} = \chi \circ \Phi$. Similarly to the divergence operator, it can be shown that for this mapping the gradient operator transforms as

$$\nabla \chi = \mathbf{J}^{-T} \hat{\nabla} \hat{\chi}. \tag{A.15}$$

On the reference cell, the commutativity relation (A.13) is then given by

$$\begin{aligned} \nabla(\pi^0\psi) = \pi^1(\nabla\psi) \quad &\rightarrow \quad \mathbf{J}^{-T}\hat{\nabla}\widehat{(\pi^0\psi)} = \mathbf{J}^{-T}\widehat{\pi^1(\nabla\psi)} \\ &\rightarrow \quad \hat{\nabla}\widehat{(\pi^0\psi)} = \widehat{\pi^1(\nabla\psi)}, \end{aligned} \tag{A.16}$$

as required, where we used (A.15) for $\chi = \pi^0\psi$ and the covariant Piola transformation for $\pi^1(\nabla\psi) \in V_h^1(\Omega)$. Finally, next to the divergence and gradient operators, commutativity on the reference cell can also be shown for the curl operator. The derivation is analogous to (A.12) and (A.16), and relies on the curl operator form for a velocity field $\mathbf{v}$ under the covariant Piola transformation:

$$\nabla \times \mathbf{v} = \frac{\mathbf{J}\hat{\nabla} \times \hat{\mathbf{v}}}{\det \mathbf{J}}. \tag{A.17}$$

Altogether, in three dimensions, the conformal and compatible cell mappings are given by

$$\eta \in V_h^0: \qquad \eta(\mathbf{x}) = \hat{\eta}(\hat{\mathbf{x}}), \tag{A.18}$$

$$\boldsymbol{\Sigma} \in V_h^1: \qquad \boldsymbol{\Sigma}(\mathbf{x}) = \mathbf{J}^{-T}\hat{\boldsymbol{\Sigma}}(\hat{\mathbf{x}}), \tag{A.19}$$

$$\mathbf{w} \in V_h^2: \qquad \mathbf{w}(\mathbf{x}) = \frac{\mathbf{J}\hat{\mathbf{w}}(\hat{\mathbf{x}})}{\det \mathbf{J}}, \tag{A.20}$$

$$\phi \in V_h^3: \qquad \phi(\mathbf{x}) = \frac{\hat{\phi}(\hat{\mathbf{x}})}{\det \mathbf{J}}, \tag{A.21}$$

and similar in 2D. The compatible finite element spaces are then constructed using these mappings together with the corresponding finite elements described in Section 1.1. Finally, we note that in this section, we assumed canonical projections π^i, which are surjective and which we further showed to be commuting. However, they may not be bounded as functions of the underlying Sobolev spaces, and further modifications including mollifiers are required to resolve this problem. For more details on this topic, see [AFW06, Arn18].

Remark 34. *The construction of compatible finite element spaces can also be described using differential forms in a framework called finite element exterior calculus. V_h^k is considered as a space of k-forms, and the corresponding cell mappings are formulated to preserve the forms. For instance, V_h^3 contains volume forms, and the division by* $\det \mathbf{J}$ *in transformation* (A.21) *ensures conservation of volume. More details on this approach can*

be found e.g. in [Arn18]. It is also worth noting that a similar framework, called discrete exterior calculus, exists and can be found e.g. in [DHLM05].

A.1 Rehabilitation

For non-affine mappings Φ (required e.g. for cubed sphere grids), the inverse of the Jacobian determinant, i.e. $1/\det \mathbf{J}$, leads to a non-polynomial term, which cannot be integrated exactly using a quadrature rule. In particular, this inverse occurs in the weak form of the divergence operator, where for $\mathbf{w} \in V_h^2(\Omega)$, $\phi \in V_h^3(\Omega)$, and $K \in \mathcal{T}$, we have

$$\int_K \phi \nabla \cdot \mathbf{w} \, d\hat{x} = \int_{\hat{K}} \frac{\hat{\phi}}{\det \mathbf{J}} \frac{\widehat{\nabla \cdot \mathbf{w}}}{\det \mathbf{J}} \det \mathbf{J} \, d\hat{x} = \int_{\hat{K}} \frac{\hat{\phi} \widehat{\nabla \cdot \mathbf{w}}}{\det \mathbf{J}} \, d\hat{x}. \tag{A.22}$$

To avoid this, we revert to a method referred to as rehabilitation [BR09], where the cell mapping (A.21) corresponding to $V_h^3(\Omega)$ is replaced by

$$\phi \in V_h^3 : \qquad \phi(\mathbf{x}) = \hat{\phi}(\hat{\mathbf{x}}). \tag{A.23}$$

In this case, however, the cell-wise commutativity relation (A.12) does not hold anymore. To recover it, the discrete divergence operator is redefined (following [BR09]) such that after computing the divergence, we additionally apply the $V_h^3(\Omega)$ space projection π^3, i.e. $\nabla\cdot \rightarrow \pi^3 \nabla\cdot$. The commutativity relation then maps to the reference cell according to

$$\pi^3 \left(\nabla \cdot (\pi^2 \mathbf{u}) \right) = \pi^3 (\nabla \cdot \mathbf{u}) \quad \rightarrow \quad \widehat{\pi^3 \left(\nabla \cdot (\pi^2 \mathbf{u}) \right)} = \widehat{\pi^3 (\nabla \cdot \mathbf{u})}, \tag{A.24}$$

noting that both $\pi^3 \left(\nabla \cdot (\pi^2 \mathbf{u}) \right)$ and $\pi^3 (\nabla \cdot \mathbf{u})$ are functions of $V_h^3(\Omega)$ and are therefore transformed according to (A.23).

To this end, we note that in the computational implementation described in the numerical results sections – see e.g. near the bottom of Section 2.2.1 – a cell mapping of the form (A.23) is used for $V_h^3(\Omega)$ in the 3D case and $V_h^2(\Omega)$ in the 2D case. However, the additional adjustment to the divergence operator, i.e. $\nabla\cdot \rightarrow \pi^3 \nabla\cdot$ (and $\nabla\cdot \rightarrow \pi^2 \nabla\cdot$ in 2D, respectively), is not included, which constitutes a deviation from the compatible framework. From the point of view of stability e.g. for the pressure gradient term in the shallow water equations – which first appear in this thesis in their linearised form (1.24) - (1.25) in Section 1.1.3 – this has no negative effect, since the pressure gradient term is

given by

$$\langle \nabla \cdot \mathbf{w}, gD \rangle, \tag{A.25}$$

for velocity test function $\mathbf{w} \in V_h^1(\Omega)$, constant gravitational acceleration g, and depth $D \in V_h^2(\Omega)$. In particular, since $gD \in V_h^2(\Omega)$, the divergence term $\nabla \cdot \mathbf{w}$ in the inner product is equal to $\pi^2(\nabla \cdot \mathbf{w})$. The stability analysis as given in [CS12] therefore applies to either definition of the divergence operator, and their is no caveat to using $\nabla \cdot \mathbf{w}$ when compared to the correct choice $\pi^2(\nabla \cdot \mathbf{w})$ required to obtain a compatible finite element complex, given the transformation (A.23). However, for other appearances of the divergence operator, which are not paired with an element of the discontinuous space, the use of $\nabla \cdot \mathbf{w}$ rather than $\pi^2(\nabla \cdot \mathbf{w})$ does introduce an additional error. While this may e.g. have a negative effect on energy conservation, which is one of this thesis' main concepts, the numerical results in Chapters 2, 3, and 4 suggest that this is not the case.

B

Hamiltonian formulation based on momentum

The standard bracket given in Definition 5 in Section 1.2 is one of many possible brackets leading to the governing equations used in this thesis. Here, we present an alternative bracket that is based on momentum instead of velocity. To do so, we make use of differential forms as well as variational fluid dynamics, and will consider the so-called Lie-Poisson bracket, following e.g. [HMR98, MR13]. Next to presenting the bracket and its relation to the Euler-Poincaré equation, the appendix also includes an observation on challenges with respect to space discretisations for this framework, thereby motivating the preference of the bracket used in this thesis over the one described here. Further, it includes an observation that sheds light on the role of the velocity recovery operator introduced in Chapter 2 and its relation to the momentum based framework.

B.1 Preliminaries

Before presenting the bracket, we briefly and informally review the underlying differential forms and Lie derivative framework. Further, we include a discussion on variational fluid dynamics, since the bracket is closely related to this approach. For a detailed description of these topics, we refer to [HMR98, MR13] as well as [Arn18] for differential forms in view of finite element spaces.

Differential forms

The language of differential forms allows for a generalised framework of calculus on manifolds. Informally, a smooth manifold Ω is a space that admits the operations of calculus in local coordinates. Its tangent space $T_x\Omega$ is given by the space of vectors tangent to Ω when passing through the point x. A differential form of degree k (in short a 'k-form') is a function $\alpha(x)\colon T_x\Omega \times ... \times T_x\Omega \to \mathbb{R}$ (for k factors $T_x\Omega$) which assigns to each point $x \in \Omega$ a skew-symmetric k-multilinear map on the tangent space $T_x\Omega$. The space of k-forms is denoted by $\Lambda^k(\Omega)$. k-forms can be expressed locally using a basis $\{dx^1, ..., dx^n\}$ of the dual space of $T_x\Omega$ as

$$\alpha(x)(v_1, ..., v_k) = \sum_{1 \leq \sigma_1 < ... < \sigma_k \leq n} \alpha_\sigma(x) dx^{\sigma_1} \wedge ... \wedge dx^{\sigma_k}, \tag{B.1}$$

where the wedge product $\wedge\colon \Lambda^j(\Omega) \times \Lambda^k(\Omega) \to \Lambda^{j+k}(\Omega)$ is given by

$$(\alpha \wedge \beta)(v_1, ..., v_{j+k}) = \sum_\sigma \operatorname{sign}(\sigma)\ \alpha(v_{\sigma_1}, ..., v_{\sigma_j}) \beta(v_{\sigma_{j+1}}, ..., v_{\sigma_{j+k}}). \tag{B.2}$$

Note that for notational convenience, we dropped all occurrences of $x \in \Omega$, and will continue to do so thereafter. For instance, if $\Omega = \mathbb{R}^n$, $\alpha \in \Lambda^1(\Omega)$, and the coefficients α_σ are specified by an element $a = (a_1, ..., a_n)$ of the space of vector fields $\mathfrak{X}(\Omega)$, then α applied to another vector field $b \in \mathfrak{X}(\Omega)$ is given by

$$\alpha(b) = \sum_{j=1}^n a_j dx^j(b) = \sum_{j=1} a_j b_j = a \cdot b, \tag{B.3}$$

since in this case the dual function evaluations $dx^j(b)$ simply return the vector components b_j. In what follows, we will also use the interior product (or "contraction") of differential forms, which maps from $\Lambda^k(\Omega)$ to $\Lambda^{k-1}(\Omega)$ by means of a vector v and is given by

$$\alpha \lrcorner v(v_1, v_2, ..., v_k) = \alpha(v, v_2, ..., v_k). \tag{B.4}$$

A fundamental operator defined on k-forms is given by the exterior derivative d, which maps to $k+1$ forms. For $\alpha \in \Lambda^k(\Omega)$ written in dual basis form (B.1), we have

$$d\alpha = \sum_\sigma \sum_{j=1}^n \frac{\partial \alpha_\sigma}{\partial x^j} dx^j \wedge dx^{\sigma_1} \wedge ... \wedge dx^{\sigma_k}. \tag{B.5}$$

Note that in particular, given the wedge product's alternating sign definition, we find that for n-forms, the exterior derivative evaluates to zero. Further, we find that $d^2\alpha = 0$ for

any k-form α.

For local coordinate systems on a given manifold, we can translate the form notation introduced above back to the usual calculus notation. For instance, for a 0-form f (noting that 0-forms can be associated with scalars), we have

$$df = \sum_{j=1}^{n} \frac{\partial f}{\partial x^j} dx^j = \nabla f \cdot dx \in \Lambda^1(\Omega), \tag{B.6}$$

for $dx = (dx^1, ..., dx^n)$. Further, for a three-dimensional manifold, and a 1-form α specified by a vector field a (as in (B.3) above), we find

$$d\alpha = d\left(\sum_{j=1}^{3} a_j dx^j\right) = \sum_{i,j} \frac{\partial a_i}{\partial x^j} dx^j \wedge dx^i = (\nabla \times a) \cdot dS \in \Lambda^2(\Omega), \tag{B.7}$$

for surface form $dS = (dx^2 \wedge dx^3, dx^3 \wedge dx^1, dx^1 \wedge dx^2)$ (noting that the dx^i themselves in turn can be seen as line forms). Using $a = \nabla f$, so that from (B.6) we have $\alpha = df$, we then recover the vector calculus identity

$$0 = d^2 f = d(df) = (\nabla \times (\nabla f)) \cdot dS. \tag{B.8}$$

In what follows below, we will further make use of the following two expressions related to the interior product, for a vector field b, 1-form $\alpha = a \cdot dx$ and so-called volume form $dV = dx^1 \wedge ... \wedge dx^n$:

$$b \lrcorner (d\alpha) = b \lrcorner \big((\nabla \times a) \cdot dS\big) = \big((\nabla \times a) \times b\big) \cdot dx \in \Lambda^1(\Omega), \tag{B.9}$$

$$d(b \lrcorner dV) = (\nabla \cdot b) dV \in \Lambda^n(\Omega). \tag{B.10}$$

Finally, we note that this framework is frequently used in the context of integration, in the sense that we integrate k-forms over subsets of Ω of dimension k. For the exterior derivative, there exists a generalised Stokes theorem: for $\alpha \in \Lambda^k(\Omega)$, and $M \subseteq \Omega$ such that the dimension of M is equal to $k + 1$, we have

$$\int_M d\alpha = \int_{\partial M} \alpha. \tag{B.11}$$

Lie derivatives

Differential forms provide a coordinate free framework for multivariable calculus. Similarly, the Lie derivative evaluates the change of a function along a vector field in a coordinate free way. In the framework of differential forms, it can be expressed for $\alpha \in \Lambda^k(\Omega)$ via Cartan's formula as

$$\pounds_u(\alpha) = u \lrcorner d\alpha + d(u \lrcorner \alpha), \tag{B.12}$$

for a vector field u. For scalar functions f (i.e. 0-forms), we obtain the usual directional derivative

$$\pounds_u(f) = u \lrcorner (\nabla f \cdot dx) + d(0) = u \cdot \nabla f, \tag{B.13}$$

where we followed (B.3), (B.6), and relied on the fact that f is a zero form (so that $u \lrcorner f = 0$). For 1-forms $\alpha = a \cdot dx$, we find

$$\pounds_u(\alpha) = d(u \lrcorner \alpha) + u \lrcorner d\alpha = \big(\nabla(u \cdot a) + (\nabla \times a) \times u\big) \cdot dx, \tag{B.14}$$

noting that $u \lrcorner \alpha = u \cdot a \in \Lambda^0(\Omega)$, and where we used (B.9). Finally, for a volume form $f dV$ with scalar function f, we have

$$\pounds_u(f dV) = d\big(u \lrcorner (f dV)\big) + u \lrcorner d(f dV) = \big(\nabla \cdot (fu)\big) dV, \tag{B.15}$$

noting that $d(f dV) = 0$ as dV is an n-form, and where we used (B.10).

Variational fluid dynamics

Having defined the Lie derivative and computed it for 0, 1, and n-forms, we move on to its application within variational fluid dynamics. In the latter, we consider variations to the action

$$S(x) = \int_0^T l(x, \dot{x}) dt, \tag{B.16}$$

for a Lagrangian l, and assume that, following the least action principle, these variations are equal to zero. For instance, for the Lagrangian corresponding to kinetic energy, the action's variation is given by

$$\delta S = \delta \int_0^T \int_\Omega \frac{1}{2} D|u|^2 \, dV \, dt = \int_0^T \int_\Omega (Du \; \delta u) dV \, dt, \tag{B.17}$$

where we assumed that the corresponding fluid is incompressible, so that D is constant. For fluid flow, it can be shown using flow charts that the velocity variation δu is specified

according to the so-called "Lin constraint" [HMR98]

$$\delta u = \partial_t w + [u, w], \tag{B.18}$$

where the Jacobi-Lie bracket $[.,.]$ is given by

$$[u, w] = (u \cdot \nabla)w - (w \cdot \nabla)u, \tag{B.19}$$

and w is an arbitrary velocity field (up to boundary conditions). With this variation, it can be shown that the least action principle implies for (B.17)

$$0 = \delta S = \int_0^T \int_\Omega w \cdot (-\partial_t u - \nabla(u \otimes u) - \nabla u^T u) dV \, dt, \tag{B.20}$$

where we applied integration by parts in t and x, and set without loss of generality $D = 1$. For incompressible flow, this then leads to the usual equations of the form

$$\partial_t u + u \cdot \nabla u + \nabla p = 0, \tag{B.21}$$

noting that the gradient term arises since

$$\int_\Omega w \cdot \nabla p \, dV = -\int_\Omega (\nabla \cdot w) p \, dV = 0, \tag{B.22}$$

as $\nabla \cdot w = 0$. In other words, (B.20) is only defined up to such a pressure gradient term.

From a differential forms point of view, we can write the variation of the Lagrangian as

$$\delta l = \left\langle \frac{\delta l}{\delta u}, \delta u \right\rangle, \tag{B.23}$$

where here $\langle ., . \rangle$ denotes pairings of elements of a given space S and elements of the corresponding dual space S^*. For instance, in (B.23) we have a vector field $\delta u \in \mathfrak{X}(\Omega)$, and a so-called 1-form density

$$\frac{\delta l}{\delta u} = (m \cdot dx) \otimes dV \in \Lambda^1(\Omega) \otimes \Lambda^n(\Omega) = \mathfrak{X}(\Omega)^*, \tag{B.24}$$

for a vector field m depending on the choice of Lagrangian. In other words, the 1-form $m \cdot dx$ is applied to the vector field δu, and the result together with dV yields an n-form

over which we then integrate:

$$\delta l = \left\langle \frac{\delta l}{\delta u}, \delta u \right\rangle = \langle (m \cdot dx) \otimes dV, \delta u \rangle = \int_\Omega m \cdot \delta u \, dV. \tag{B.25}$$

In this form, the least action principle reads

$$0 = \delta S = \int_0^T \left\langle \frac{\delta l}{\delta u}, \delta u \right\rangle dt = \int_0^T \left\langle \frac{\delta l}{\delta u}, \partial_t w + [u, w] \right\rangle dt \tag{B.26}$$

$$= \int_0^T \left(- \left\langle \partial_t \left(\frac{\delta l}{\delta u} \right), w \right\rangle + \left\langle \frac{\delta l}{\delta u}, [u, w] \right\rangle \right) dt, \tag{B.27}$$

where we applied integration by parts in t. Note that before we moved to the language of differential forms, we derived from the least action principle a transport equation of the form (B.21). Accordingly, we expect the differential form version (B.27) to also lead to a transport equation. For this purpose, we aim to express the second term in (B.27) in the form of a Lie derivative. This requires us to establish a connection between the latter and the Jacobi-Lie bracket $[.,.]$, for which we make use of the following identity:

$$[u, w] \lrcorner \frac{\delta l}{\delta u} = \pounds_u \left(w \lrcorner \frac{\delta l}{\delta u} \right) - w \lrcorner \left(\pounds_u \frac{\delta l}{\delta u} \right). \tag{B.28}$$

Note that the interior product applied to 1-form densities returns a volume form, i.e.

$$[u, w] \lrcorner \frac{\delta l}{\delta u} = [u, w] \lrcorner (m \cdot dx) dV = ([u, w] \cdot m) dV, \tag{B.29}$$

and similarly for $w \lrcorner \frac{\delta l}{\delta u}$. Further, note that in the first term on the right-hand side of (B.28), the Lie derivative is applied to a volume form and therefore returns a volume form, while in the second term the Lie derivative is applied to a 1-form density and so returns a 1-form density. For the first term, we use Cartan's formula, noting that since $w \lrcorner \frac{\delta l}{\delta u}$ is a volume form its exterior derivative vanishes. This leaves us with

$$\pounds_u \left(w \lrcorner \frac{\delta l}{\delta u} \right) = d\Big(u \lrcorner \big(w \lrcorner \left((m \cdot dx) \otimes dV \right) \big) \Big). \tag{B.30}$$

Applying the generalised Stokes theorem and assuming free-slip boundary conditions, we find

$$\int_\Omega \pounds_u \left(w \lrcorner \frac{\delta l}{\delta u} \right) = \int_{\partial\Omega} u \lrcorner \big(w \lrcorner (m \cdot dx) \otimes dV \big) \tag{B.31}$$

$$= \int_{\partial\Omega} u \lrcorner (w \cdot m \; dV) \tag{B.32}$$

$$= \int_{\partial\Omega} (w \cdot m) u \cdot dS = 0, \tag{B.33}$$

noting that $u \cdot dS = (u \cdot n) ds$, for boundary outward unit normal n. Altogether, only the second term in (B.28) remains, i.e.

$$[u, w] \lrcorner \frac{\delta l}{\delta u} = -w \lrcorner \left(\pounds_u \frac{\delta l}{\delta u} \right). \tag{B.34}$$

Integrating over Ω, this leads to

$$\left\langle \frac{\delta l}{\delta u}, [u, w] \right\rangle = - \left\langle w, \left(\pounds_u \frac{\delta l}{\delta u} \right) \right\rangle, \tag{B.35}$$

that is we can see the negative Lie derivative $-\pounds_u(\,.\,)$ as the adjoint of the Lie bracket operator $[u,\,.\,]$. Writing

$$ad_u(\,.\,) \coloneqq [u,\,.\,], \qquad\qquad ad^*_u(\,.\,) = -\pounds_u(\,.\,), \tag{B.36}$$

we can then reformulate (B.27) to

$$0 = \int_0^T - \left\langle w, (\partial_t - ad^*_u) \left(\frac{\delta l}{\delta u} \right) \right\rangle dt. \tag{B.37}$$

Since this holds for any w, we arrive at the definition below, noting that the pressure gradient term follows as before.

Definition 35. *Consider a Lagrangian $l = \frac{1}{2} \int_\Omega u \cdot m \; dV$, for fluid velocity u and momentum $m = uD$. Then the Euler-Poincaré equation is given by*

$$\partial_t \big(m \cdot dx \otimes dV \big) - ad^*_u \big(m \cdot dx \otimes dV \big) + dp = 0. \tag{B.38}$$

This formulation gives rise to various systems of PDEs, depending on the choice of Lagrangian.

Before moving on to a Poisson bracket framework that is analogous to the Euler-Poincaré

setup, we extend the latter to include advected quantities $a_1, ...a_m$, with a Lagrangian given by $l = l(u, a_1, ..., a_m)$. In this case, its variation extends to

$$\delta l = \left\langle \frac{\delta l}{\delta u}, \delta u \right\rangle + \sum_{i=1}^{m} \left\langle \frac{\delta l}{\delta a_i}, \delta a_i \right\rangle, \tag{B.39}$$

where the form of $\frac{\delta l}{\delta a_i}$ depends on the advected quantity (e.g. if a_i is a volume form, then $\frac{\delta l}{\delta a_i}$ is a scalar). As with the Lin constraint type variations (B.18), the variations for advected quantities follow a certain form for fluids, and it can be shown that this form is determined by

$$\delta a_i + \pounds_w(a_i) = 0. \tag{B.40}$$

Using this Lagrangian, the least action principle evaluates to

$$0 = \int_0^T \left(\left\langle \frac{\delta l}{\delta u}, \delta u \right\rangle + \sum_{i=1}^{m} \left\langle \frac{\delta l}{\delta a_i}, \delta a_i \right\rangle \right) dt \tag{B.41}$$

$$= \int_0^T \left(- \left\langle w, (\partial_t - ad_u^*) \left(\frac{\delta l}{\delta u} \right) \right\rangle - \sum_{i=1}^{m} \left\langle \frac{\delta l}{\delta a_i}, \pounds_w(a_i) \right\rangle \right) dt, \tag{B.42}$$

where the variation in u was treated as above, and for the variations in a_i we used (B.40). To arrive at a strong equation similar to (B.38), the so-called diamond operator needs to be introduced. For a space S of advected quantities (with dual space S^*), it is given by

$$\diamond \colon S^* \times S \ \to \ \Lambda^1(\Omega) \otimes \Lambda^n(\Omega) \tag{B.43}$$

such that

$$\langle b \diamond a, w \rangle = -\langle b, \pounds_w(a) \rangle, \tag{B.44}$$

for any $w \in \mathfrak{X}(\Omega)$. From (B.42), we then obtain the Euler-Poincaré equation with advected quantities, given in the definition below.

Definition 36. *For Lagrangian $l(u, a_1, ..., a_m)$ with flow velocity u and advected quantities $a_1, ..., a_m$, the Euler-Poincaré equation is given by*

$$\partial_t \left(\frac{\delta l}{\delta u} \right) - ad_u^* \left(\frac{\delta l}{\delta u} \right) = \sum_{i=1}^{m} \frac{\delta l}{\delta a_i} \diamond a_i. \tag{B.45}$$

Additionally, each advected quantity is governed by an equation of the form

$$\partial_t a_i + \pounds_u(a_i) = 0, \qquad\qquad i = 1, ..., m. \tag{B.46}$$

Note that as for the case without advected quantities, (B.45) is an equation for 1-form densities. Further, we implicitly assume the density related field to be one of the advected quantities, so that the flow is compressible and there is no pressure term. Additionally, the transport equations (B.46) can be derived in a similar manner to the variations δa_i.

Rotating shallow water equations

To illustrate the computation of governing equations using the Euler-Poincaré equations with advected quantities, we consider the Lagrangian corresponding to the rotating shallow water equations, which is given by

$$l(u, DdV) = \int_\Omega \left(\frac{1}{2}|u|^2 + R \cdot u - \frac{1}{2} g D \right) DdV, \tag{B.47}$$

for water depth D, gravitational acceleration parameter g, and R such that $\nabla \times R = 2\boldsymbol{\Omega}$, for rotation vector $\boldsymbol{\Omega}$. We then find

$$\frac{\delta l}{\delta u} = D(u + R) \cdot dx \otimes dV \ \in \Lambda^1(\Omega) \otimes \Lambda^n(\Omega), \tag{B.48}$$

$$\frac{\delta l}{\delta (DdV)} = \frac{1}{2}|u|^2 + R \cdot u - gD \quad \in \Lambda^0(\Omega). \tag{B.49}$$

First, note that from the transport equation (B.46) with $a_i = DdV$, we obtain the continuity equation:

$$\partial_t (DdV) + \pounds_u (DdV) = 0 \quad \Longrightarrow \quad D_t + \nabla \cdot (uD) = 0, \tag{B.50}$$

where we used (B.15) to evaluate the Lie derivative, and further that the volume form dV is independent of time. To derive the momentum equation, we write $m = D(u + R)$ (in line with the notation for (B.24) above), and consider the Euler-Poincaré equation with a vector field $w \in \mathfrak{X}(\Omega)$, i.e.

$$\langle w, m_t \cdot dx \otimes dV \rangle + \langle w, \pounds_u (m \cdot dx \otimes dV) \rangle = \left\langle w, \frac{\delta l}{\delta D dV} \diamond DdV \right\rangle. \tag{B.51}$$

For the Lie derivative term on the left-hand side, we apply the product rule to obtain

$$\langle w, \pounds_u (m \cdot dx \otimes dV) \rangle = \left\langle w, \pounds_u \left(\frac{m}{D} \cdot dx \right) \otimes DdV \right\rangle + \left\langle w, \left(\frac{m}{D} \cdot dx \right) \otimes \pounds_u (DdV) \right\rangle \tag{B.52}$$

$$= \left\langle w, \left(\nabla \left(u \cdot \frac{m}{D} \right) + \left(\nabla \times \frac{m}{D} \right) \times u \right) \cdot dx \otimes DdV \right\rangle + \left\langle w, \left(\frac{m}{D} \cdot dx \right) \otimes \pounds_u (DdV) \right\rangle, \tag{B.53}$$

where we used (B.14) to evaluate the Lie derivative. Further, we can reformulate the time-derivative term to

$$\langle w, m_t \cdot dx \otimes dV\rangle = \langle w, \left(D\,\frac{m}{D}\right)_t \cdot dx \otimes dV\rangle \tag{B.54}$$

$$= \left\langle Dw, \left(\frac{m}{D}\right)_t \cdot dx \otimes dV \right\rangle + \left\langle w, \left(\frac{m}{D} \cdot dx\right) \otimes D_t\, dV \right\rangle. \tag{B.55}$$

We can then use the continuity equation $(DdV)_t = -\pounds_u(DdV)$, so that the last terms in (B.53) and (B.55) cancel. Finally, for the right-hand side term in (B.51), we apply the definition of the diamond operator to obtain

$$\left\langle w, \frac{\delta l}{\delta D dV} \diamond DdV \right\rangle = -\left\langle \frac{\delta l}{\delta D dV}, \pounds_w(DdV) \right\rangle = -\left\langle \frac{\delta l}{\delta D dV}, \nabla \cdot (Dw) dV \right\rangle. \tag{B.56}$$

Assuming a constant in time Coriolis potential R (so that $(m/D)_t = u_t$) further yields

$$\left\langle Dw, \left(u_t + \left(\nabla\left(u \cdot \tfrac{m}{D}\right) + \left(\nabla \times \tfrac{m}{D}\right) \times u\right)\right) \cdot dx \otimes dV \right\rangle = -\left\langle \tfrac{\delta l}{\delta D dV}, \nabla \cdot (Dw) dV \right\rangle. \tag{B.57}$$

Note that using $\nabla \times (m/D) = \nabla \times u + 2\boldsymbol{\Omega}$, we can reformulate the advection and Coriolis terms to

$$\nabla\left(u \cdot \frac{m}{D}\right) + \left(\nabla \times \frac{m}{D}\right) \times u = \nabla\left(u \cdot (u + R)\right) + (\nabla \times u) \times u + 2\boldsymbol{\Omega} \times u. \tag{B.58}$$

Finally, applying integration by parts to the right-hand side of (B.57), and inserting the expression for $\frac{\delta l}{\delta D dV}$, we arrive at

$$\int_\Omega Dw \cdot \left(u_t + (\nabla \times u) \times u + 2\boldsymbol{\Omega} \times u + \nabla\left(\frac{1}{2}|u|^2 + gD\right)\right) dV = 0, \tag{B.59}$$

noting that we combined the first term on the right-hand side of (B.58) with the corresponding term in $\frac{\delta l}{\delta D dV}$. This corresponds to the usual vector invariant form of the momentum equation.

B.2 Lie-Poisson bracket formulation

In this section, we present a momentum based Poisson bracket that can be seen as analogous to the Euler-Poincaré equation with advected quantities, as described above [HMR98]. For this purpose, we write the 1-form density (B.24) as

$$\mu \coloneqq (m \cdot dx) \otimes dV. \tag{B.60}$$

Note that for a Lagrangian that includes the total kinetic energy

$$\frac{1}{2}\int_\Omega D|u|^2 dV, \tag{B.61}$$

m will include a contribution of the form uD, so that m – and therefore μ – is associated with momentum. μ is then used in a momentum based bracket as defined below. The Hamiltonian H can be expressed using μ and the Lagrangian l via the so-called Legendre transformation as

$$H(\mu, a_1, ..., a_m) = \langle \mu, u\rangle - l(u, a_1, ..., a_m), \tag{B.62}$$

where the velocity u is considered as a function of $(\mu, a_1, ..., a_m)$. This choice of Hamiltonian yields variations of the form

$$\frac{\delta H}{\delta \mu} = u + \left\langle \mu, \frac{\delta u}{\delta \mu}\right\rangle - \left\langle \frac{\delta l}{\delta u}, \frac{\delta u}{\delta \mu}\right\rangle = u, \tag{B.63}$$

$$\frac{\delta H}{\delta a_i} = -\frac{\delta l}{\delta a_i}, \qquad i = 1, ...m, \tag{B.64}$$

noting that $\mu = \frac{\delta l}{\delta u}$.

Definition 37. *For any functionals F, H of the prognostic variables $(\mu, a_1, ..., a_m)$, the* ***Lie-Poisson bracket*** *$\{F, H\} \in \mathbb{R}$ is given by*

$$\{F, H\} \coloneqq -\left\langle \mu, \left[\frac{\delta F}{\delta \mu}, \frac{\delta H}{\delta \mu}\right]\right\rangle + \sum_{i=1}^{m}\left(\left\langle \pounds_{\frac{\delta F}{\delta \mu}}(a_i), \frac{\delta H}{\delta a_i}\right\rangle - \left\langle \pounds_{\frac{\delta H}{\delta \mu}}(a_i), \frac{\delta F}{\delta a_i}\right\rangle\right). \tag{B.65}$$

Recall that the bracket $\langle ., .\rangle$ is considered for parings of spaces S and their dual S^*. For instance, in the first term of (B.65), we find that $[., .]$ is a vector field, and μ is a 1-form density, which leads to

$$\left\langle \mu, \left[\frac{\delta F}{\delta \mu}, \frac{\delta H}{\delta \mu}\right]\right\rangle = \left\langle (m \cdot dx) \otimes dV, \left[\frac{\delta F}{\delta \mu}, \frac{\delta H}{\delta \mu}\right]\right\rangle = \int_\Omega m \cdot \left[\frac{\delta F}{\delta \mu}, \frac{\delta H}{\delta \mu}\right] \, dV. \tag{B.66}$$

Further, note that $\{., .\}$ is clearly bilinear and antisymmetric since the Jacobi-Lie bracket $[., .]$ is. It can additionally be shown that it satisfies the Jacobi identity, so that the Lie-Poisson bracket is indeed a Poisson bracket as defined in Section 1.2. Given (B.65), we can proceed as before by considering the Poisson system

$$\frac{dF}{dt} = \{F, H\}. \tag{B.67}$$

Theorem 38. *([HMR98]) The Euler-Poincaré equation with advected quantities as given*

in Definition 36 is equivalent to the Poisson system (B.67) *together with the Lie-Poisson bracket as given in Definition 37.*

Proof. *For a functional of the form $F = \langle \mu, w \rangle$ with $w \in \mathfrak{X}(\Omega)$, we obtain a momentum equation given by*

$$\partial_t \langle \mu, w \rangle = - \left\langle \mu, \left[w, \frac{\delta H}{\delta \mu} \right] \right\rangle + \sum_{i=1}^{m} \left\langle \pounds_w(a_i), \frac{\delta H}{\delta a_i} \right\rangle \tag{B.68}$$

$$= + \langle \mu, [u, w] \rangle - \sum_{i=1}^{m} \left\langle \pounds_w(a_i), \frac{\delta l}{\delta a_i} \right\rangle \tag{B.69}$$

$$= \langle ad^*_u(\mu), w \rangle + \sum_{i=1}^{m} \left\langle w, \frac{\delta l}{\delta a_i} \diamond a_i \right\rangle . \tag{B.70}$$

Since this holds for any vector field w (with suitable boundary conditions), this implies the Euler-Poincaré equation as given in Definition 36. Similarly, we obtain the transport equations (B.46) *by setting $F = \langle a_i, b \rangle$, where b is any given function in the dual space of the space to which a_i belongs:*

$$\partial_t \langle a_i, b \rangle = - \left\langle \pounds_{\frac{\delta H}{\delta \mu}}(a_i), b \right\rangle = - \langle \pounds_u(a_i), b \rangle \tag{B.71}$$

$$\implies \langle b, \partial_t a_i + \pounds_u(a_i) \rangle = 0. \tag{B.72}$$

Conversely, for any functional $F(\mu, a_1, ..., a_m)$, we have

$$\frac{dF}{dt} = \left\langle \frac{\delta F}{\delta \mu}, \frac{d\mu}{dt} \right\rangle + \sum_{i=1}^{m} \left\langle \frac{\delta F}{\delta a_i}, \frac{da_i}{dt} \right\rangle \tag{B.73}$$

$$= \left\langle \frac{\delta F}{\delta \mu}, ad^*_u(\mu) \right\rangle + \sum_{i=1}^{m} \left\langle \frac{\delta F}{\delta \mu}, \frac{\delta l}{\delta a_i} \diamond a_i \right\rangle - \sum_{i=1}^{m} \left\langle \frac{\delta F}{\delta a_i}, \pounds_u(a_i) \right\rangle , \tag{B.74}$$

$$= \left\langle ad^*_{\frac{\delta H}{\delta \mu}}(\mu), \frac{\delta F}{\delta \mu} \right\rangle + \sum_{i=1}^{m} \left(- \left\langle \pounds_{\frac{\delta F}{\delta \mu}}(a_i), \frac{\delta l}{\delta a_i} \right\rangle - \left\langle \frac{\delta F}{\delta a_i}, \pounds_u(a_i) \right\rangle \right) , \tag{B.75}$$

$$= - \left\langle \mu, \left[\frac{\delta F}{\delta \mu}, \frac{\delta H}{\delta \mu} \right] \right\rangle + \sum_{i=1}^{m} \left(\left\langle \pounds_{\frac{\delta F}{\delta \mu}}(a_i), \frac{\delta H}{\delta a_i} \right\rangle - \left\langle \pounds_{\frac{\delta H}{\delta \mu}}(a_i), \frac{\delta F}{\delta a_i} \right\rangle \right) = \{F, H\}, \tag{B.76}$$

as required. □

Note that in particular, this implies that governing equations derived from the Euler-Poincaré equation with advected quantities are energy conserving, which follows directly

from the Lie-Poisson bracket's antisymmetry. We conclude this appendix with two comments on observations regarding the space discretisation of this framework. The first one raises difficulties with such a discretisation, while the second one highlights a similarity to the framework considered in the thesis' main chapters.

Discretisation of the Lie-Poisson bracket

The Euler-Poincaré equation with advected quantities, as given in Definition 36, is an equation for 1-form densities. At the bottom of Section B.1, we used it to derive the momentum equation for the rotating shallow water equations, which we formulated as an equation for the vector field $u \in \mathfrak{X}(\Omega)$. Recall that for this purpose, we considered the product rule for the Lie derivative applied to the 1-form density, i.e.

$$\langle w, \pounds_u(m \cdot dx \otimes dV)\rangle = \left\langle w, \pounds_u\left(\frac{m}{D} \cdot dx\right) \otimes DdV\right\rangle + \left\langle w, \left(\frac{m}{D} \cdot dx\right) \otimes \pounds_u(DdV)\right\rangle, \quad \text{(B.77)}$$

and the second term on the right-hand side was then cancelled with another term emerging from the product rule applied to the time derivative, i.e.

$$\langle w, m_t \cdot dx \otimes dV\rangle = \left\langle Dw, \left(\frac{m}{D}\right)_t \cdot dx \otimes dV\right\rangle + \left\langle w, \left(\frac{m}{D} \cdot dx\right) \otimes D_t\, dV\right\rangle. \quad \text{(B.78)}$$

In other words, adding (B.77) and (B.78) together, we obtain a contribution of form

$$\left\langle \frac{w \cdot m}{D},\ D_t dV + \pounds_u(DdV)\right\rangle = \left\langle \frac{w \cdot m}{D},\ \Big(D_t + \nabla \cdot (uD)\Big)dV\right\rangle, \quad \text{(B.79)}$$

which cancels since the second term in the pairing is equal to zero due to the continuity equation. In view of finite element space discretisations, note that for the cancellation to hold true, it is sufficient for the left-hand side of the continuity equation to be equal to zero in L^2, i.e.

$$\|D_t + \nabla \cdot (uD)\|_2 = 0. \quad \text{(B.80)}$$

In terms of a derivation via the Poisson system using the Lie-Poisson bracket (B.65), the continuity equation is obtained by choosing $F = \langle \psi, DdV\rangle$, for scalar function ψ:

$$\langle \psi, (DdV)_t\rangle = -\left\langle \psi, \pounds_{\frac{\delta H}{\delta \mu}}(DdV)\right\rangle, \quad \text{(B.81)}$$

$$\implies \int_\Omega \psi D_t\, dV = -\int_\Omega \psi \nabla \cdot \left(\frac{\delta H}{\delta \mu} D\right) dV. \quad \text{(B.82)}$$

When discretising this framework using the compatible finite element spaces reviewed in Chapter 1, the term $(w \cdot m)/D$ in (B.79) will generally not a be function of the finite element depth space $\mathbb{V}_D$. In particular, for the cancellation of (B.79) to still work after discretisation, the corresponding discretised version of the continuity equation (B.82) needs to hold as an L^2 equivalence as described above. This task is nontrivial, since the term $\frac{\delta H}{\delta \mu} D$ will not generally be a function of the velocity space $\mathbb{V}_u$. In particular, in this case the divergence operator does not necessarily map the latter term into $\mathbb{V}_D$. An example of a discretisation which does not fulfil this property is e.g. given by the upwind stabilised form (see Section 1.3.1)

$$\int_\Omega \phi D_t \, dV = \int_\Omega \nabla \phi \cdot \left(\frac{\delta H}{\delta \mu} D \right) dV - \int_\Gamma \left[\!\left[\phi \frac{\delta H}{\delta \mu} \right]\!\right] \tilde{D} \, dS, \tag{B.83}$$

for test functions $\phi \in \mathbb{V}_D$. Instead, we could consider a discretisation of the advection term of the form

$$\int_\Omega \phi \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta H}{\delta \mu} D \right) dV, \tag{B.84}$$

for projection $P_{\mathbb{V}_u}$ into the velocity space. In this case, the discretised continuity equation is given by

$$\int_\Omega \phi \left(D_t + \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta H}{\delta \mu} D \right) \right) dV = 0 \qquad\qquad \forall \phi \in \mathbb{V}_D, \tag{B.85}$$

and in particular, we can pick $\phi = D_t + \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta H}{\delta \mu} D \right) \in \mathbb{V}_D$ to show that the left-hand side term $D_t + \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta H}{\delta \mu} D \right)$ in (B.85) is indeed equal to zero in L^2. The discretised version of (B.79) then satisifies

$$\left\langle \frac{w \cdot m}{D}, \, \left(D_t + \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta H}{\delta \mu} D \right) \right) dV \right\rangle = 0, \tag{B.86}$$

as required. However, in this case, the corresponding antisymmetric term arising from the Lie-Poisson bracket would be given by

$$\int_\Omega \frac{\delta H}{\delta D dV} \nabla \cdot P_{\mathbb{V}_u} \left(\frac{\delta F}{\delta \mu} D \right) dV = - \int_\Omega \frac{\delta l}{\delta D dV} \nabla \cdot P_{\mathbb{V}_u} \left(w D \right) dV, \tag{B.87}$$

which appears in the momentum equation, for test function $w \in \mathbb{V}_u$. The projection on the weighted test function wD would then lead to further complications that are nontrivial to resolve. Due to this difficulty related to cancellations of discrete terms arising from the product rule – assuming that the product rule still holds after discretisation in the first place – we did not consider a discretised Lie-Poisson bracket in this thesis. Note that the

cancellation occurs in the above framework when choosing a functional of either form

$$F = \langle w, \mu \cdot dx \otimes dV \rangle, \qquad F = \langle w, u(\mu, D) \cdot dx \otimes dV \rangle, \tag{B.88}$$

to obtain an evolution equation for momentum or velocity. In particular, the problem of satisfying this cancellation after discretisation also occurs for both choices of functionals and therefore evolution equations.

Relationship to the flux recovery operator $\mathbb{U}$

The velocity equation (B.59) resembles the setup of Chapter 2 in the sense that we arrive at weighted test functions of the form Dw. This can be seen as a consequence of the fact that the continuous variation in momentum used here is analogous to the continuous recovered variation in velocity used in Chapter 2, i.e.

$$\frac{\delta H}{\delta \mu} = u = \mathbb{U}\left(D, \frac{\delta H}{\delta u}\right), \tag{B.89}$$

up to the Coriolis term R included in the Hamiltonian (B.62) via the Lagrangian (B.47). This analogy also holds for the choice of functional $F = \int_\Omega u \cdot w \; dV$ used to obtain the velocity equation in Chapter 2, since we then find for the continuous case,

$$\mathbb{U}\left(D, \frac{\delta F}{\delta u}\right) = \frac{w}{D}, \tag{B.90}$$

and in the above framework the choice of

$$F = \int_\Omega u \cdot w \; dV = \langle w, u \cdot dx \otimes dV \rangle = \left\langle \frac{w}{D}, \mu \right\rangle, \tag{B.91}$$

also yields

$$\frac{\delta F}{\delta \mu} = \frac{w}{D} = \mathbb{U}\left(D, \frac{\delta F}{\delta u}\right). \tag{B.92}$$

The velocity recovery operator $\mathbb{U}$ can therefore be seen as a translation between the momentum and velocity frameworks. Note, however, that the Lie-Poisson bracket (B.65) and Hamiltonian (B.62) are formulated differently to the standard bracket (1.57) and Hamiltonian (1.65) in Section 1.2. In particular, as elaborated in the first observation above, the Lie-Poisson setup requires a cancellation of terms in order to lead to a velocity evolution equation. This feature is intrinsically related to the fact that momentum depends on both velocity and depth. Importantly, such a cancellation (and therefore a continuity equation that evaluates to zero in L^2) is not required in the setup in Section 1.2, where

we use variations in velocity rather than momentum. Finally, unlike the standard bracket in Section 1.2, the Lie-Poisson bracket does not naturally contain vorticity, and cannot be associated as easily with a vorticity equation as exploited in Chapter 4.

C

Illustration of upwind stabilisation

In this appendix, we review the stabilising effect of the classical DG upwind and SUPG methods used in this thesis' main chapters. In order to do so, we first shed light on the origin of the spurious oscillations that are attenuated by these methods. With this in mind, we then first move on to the classical DG upwind method, followed by the SUPG method. The section on the latter method also contains a discussion on possible choices of stabilisation parameter, including the ones used in Chapters 3 and 4.

C.1 Motivation

To demonstrate the existence of spurious oscillations in the presence of an advection term, we consider the advection-diffusion equation and follow [ESW14, QSS10]. First, we derive an a priori error estimate, and then give a simple example with oscillatory behaviour. The prognostic variable is given by a scalar quantity q, defined on a domain Ω with Dirichlet and Neumann boundary conditions given by

$$\Omega = \Omega_D \cup \Omega_N, \quad \Omega_D \cap \Omega_N = \emptyset; \tag{C.1}$$

$$q = g_D \text{ on } \Omega_D, \quad \frac{\partial q}{\partial n} = g_N \text{ on } \Omega_N. \tag{C.2}$$

The advection-diffusion equation for q is given in its weak form by

$$a(q,p) \coloneqq \langle \mathbf{u} \cdot \nabla q, p \rangle + \epsilon \langle \nabla p, \nabla q \rangle = \langle f, p \rangle + \epsilon \int_{\partial\Omega_N} g_N p \, dS =: l(p) \quad \forall p \in H^1_{E_0}(\Omega), \quad \text{(C.3)}$$

for advecting velocity $\mathbf{u}$, diffusion coefficient ϵ, and forcing term f. Further, $H^1_{E_0}(\Omega)$ is given by the space of functions in $H^1(\Omega)$ that vanish on Ω_D. Finally, we specify q to be an element of $H^1_E(\Omega)$, the space of functions in $H^1(\Omega)$ equal to g_D on Ω_D. In the discretised version, we look for q_h in a finite element space $V_E \subset H^1_E(\Omega)$ such that

$$a(q_h, p_h) = l(p_h) \qquad \forall p_h \in V_{E_0}(\Omega), \quad \text{(C.4)}$$

for finite element space $V_{E_0}(\Omega) \subset H^1_{E_0}(\Omega)$.

If we have outflow Neumann boundary conditions, i.e. $\mathbf{u} \cdot \mathbf{n} > 0$ along Ω_N (for outward normal unit vector $\mathbf{n}$), it can be shown that the bilinear form a is coercive, i.e.

$$a(q,q) \geq \epsilon \|\nabla q\|_2^2, \quad \text{(C.5)}$$

for L^2 norm $\|\cdot\|_2$. Further, using the Cauchy-Schwarz inequality and Poincaré's inequality, we have

$$|a(q,p)| \leq \epsilon |\langle \nabla q, \nabla p \rangle| + |\langle \mathbf{u} \cdot \nabla q, p \rangle| \quad \text{(C.6)}$$

$$\leq (\epsilon + C\|\mathbf{u}\|_\infty) \|\nabla q\|_2 \|\nabla p\|_2, \quad \text{(C.7)}$$

for Poincaré constant $C \geq 0$ related to the domain's size and geometry, and $\|\mathbf{u}\|_\infty = \sup_{\mathbf{x}\in\Omega} |\mathbf{u}(\mathbf{x})|$ (assuming a continuous flow $\mathbf{u}$). We can then use these two inequalities for an a priori error bound. For this purpose, we start with Galerkin orthogonality, given by

$$a(q - q_h, p_h) = 0 \qquad \forall p_h \in V_{E_0}, \quad \text{(C.8)}$$

where we used (C.3) and (C.4) with a test function p_h. Using this together with the above two inequalities, we then have for $r_h \in V_E$,

$$\epsilon \|\nabla(q - q_h)\|_2^2 \leq a(q - q_h, q - q_h) \quad \text{(C.9)}$$

$$= a(q - q_h, q - r_h) + a(q - q_h, r_h - q_h) \quad \text{(C.10)}$$

$$= a(q - q_h, q - r_h) \quad \text{(C.11)}$$

$$\leq (\epsilon + C\|\mathbf{u}\|_\infty)\|\nabla(q - q_h)\|_2\|\nabla(q - r_h)\|_2, \tag{C.12}$$

noting that $r_h - q_h \in V_{E_0}$. Taking the infimum over all functions $r_h \in V_E$, this standard argument leads to a best approximation property of the form

$$\|\nabla(q - q_h)\|_2 \leq (1 + \mathcal{P}) \inf_{r_h \in V_E} \|\nabla(q - r_h)\|_2, \tag{C.13}$$

for the so-called Péclet number $\mathcal{P} = C\|\mathbf{u}\|_\infty/\epsilon$. In particular, we find that for convection dominated problems, where $\|\mathbf{u}\|_\infty \gg \epsilon$, the error estimate deteriorates.

This lack of accuracy typically manifests itself in the form of spurious oscillations [ESW14], and to illustrate this, we consider an example as described in [QSS10]. The domain is given by the unit interval, with Dirichlet boundary conditions $q(0) = 0,\ q(1) = 1$. Further, we assume a constant advecting velocity u and zero forcing $f = 0$, and consider V_E to be CG_1 on a uniform mesh (i.e. a mesh with cells of equal shape and size). Expanding q in the element basis of V_E as

$$q = \sum_{k=0}^{N} \hat{q}_k p_k, \tag{C.14}$$

for coefficients $\hat{q}_k$ and $N + 1$ basis functions $p_k \in V_E$, it can be shown that the weak form (C.4) evaluates to

$$\left(\frac{u}{2} - \frac{\epsilon}{h}\right)\hat{q}_{k+1} + \frac{2\epsilon}{h}\hat{q}_k - \left(\frac{u}{2} + \frac{\epsilon}{h}\right)\hat{q}_{k-1} = 0 \qquad k = 1, ...N - 1, \tag{C.15}$$

for uniform cell length $h = 1/N$. Using the so-called element Péclet number $\mathcal{P}_h = \|u\|_2 h/2\epsilon$, the coefficients can be solved for as

$$\hat{q}_k = \frac{\alpha^k - 1}{\alpha^N - 1}\,, \qquad \alpha := \frac{1 + \mathcal{P}_h}{1 - \mathcal{P}_h}. \tag{C.16}$$

For $\mathcal{P}_h > 1$, we have $\alpha < 0$, so that its k^{th} power in the numerator of $\hat{q}_k$ changes sign for varying values of k. In particular, in line with the error bound above, this may lead to oscillations in the solution coefficients $\{\hat{q}_k\}_{k=0}^N$. As $\mathcal{P}_h$ tends to infinity, α tends to -1, leading to a numerator that alternates between 0 and -2 as k varies from 0 to N. In this case, we therefore arrive at a solution that is dominated by oscillations throughout the domain. Note that if there is no advection term (i.e. $u = 0$) the solution (C.15) reduces to

$$-\hat{q}_{k+1} + 2\hat{q}_k - \hat{q}_{k-1} = 0 \qquad k = 1, ...N - 1. \tag{C.17}$$

This then evaluates to

$$\hat{q}_k = \frac{k}{N}, \tag{C.18}$$

which is free of oscillations.

In practice, such oscillations are countered by discretisations that can be interpreted as adding artificial diffusion, which can be achieved in many ways depending on the choice of finite element discretisation [MKM+16]. Indeed, the oscillatory behaviour above can be seen as a discretisation that is under-diffusive [BH82], and the addition of artificial diffusion simply removes this effect. To further justify the addition of artificial diffusion, the latter has also been interpreted from a sub-grid model point of view [BMR98]. This is consistent with (C.16), which ceases to be oscillatory for a sufficiently small cell size h, indicating that all scales are resolved, while for larger h a sub-grid model is needed. Note that for smaller values of ϵ, an ever finer resolution h would be required to ensure $\mathcal{P}_h \leq 1$. Since in the case of atmosphere and ocean modelling this would become prohibitively expensive, this interpretation implies that artificial diffusion or other stabilisation methods is a necessity [MKM+16].

The two stabilisation methods reviewed below are based on adding artificial diffusion. Since these lead to formulations that require information of the velocity from the direction the velocity is coming from, these methods are referred to as *upwind* stabilisation methods.

C.2 Classical DG upwind method

To illustrate that the classical DG upwind method, as defined in Section 1.3.1, has a diffusive effect, we consider the motivating example above and follow the discussion given in [QSS10]. After dividing by h, the discretised scheme (C.15) can be reformulated to

$$-\epsilon \frac{\hat{q}_{k+1} - 2\hat{q}_k + \hat{q}_{k-1}}{h^2} + u \frac{\hat{q}_{k+1} - \hat{q}_{k-1}}{2h} = 0 \qquad k = 1, ...N-1, \tag{C.19}$$

which is equal to the scheme resulting from a central finite difference discretisation. To avoid oscillations originating from the advection term, we add a discrete diffusive contribution of the form

$$-u\frac{h}{2}\frac{\hat{q}_{k+1} - 2\hat{q}_k + \hat{q}_{k-1}}{h^2}, \tag{C.20}$$

which leads to a new discretisation given by

$$-\epsilon\frac{\hat{q}_{k+1}-2\hat{q}_k+\hat{q}_{k-1}}{h^2}+u\frac{\hat{q}_{k+1}-\hat{q}_{k-1}}{2h}-u\frac{h}{2}\frac{\hat{q}_{k+1}-2\hat{q}_k+\hat{q}_{k-1}}{h^2}=0, \tag{C.21}$$

$$\implies \quad -\epsilon\frac{\hat{q}_{k+1}-2\hat{q}_k+\hat{q}_{k-1}}{h^2}+u\frac{\hat{q}_k-\hat{q}_{k-1}}{h}=0, \qquad k=1,...N-1. \tag{C.22}$$

In other words, the addition of numerical diffusion leads to a discretisation of the advection term which considers the current location $\hat{q}_k$ together with the location in the upwind direction (i.e. $\hat{q}_{k-1}$ for $u \geq 0$, and similarly $\hat{q}_{k+1}$ for $u < 0$). Note that in the limit as ϵ tends to zero, the central difference scheme (C.19) reduces to

$$u\frac{\hat{q}_{k+1}-\hat{q}_{k-1}}{2h}=0 \qquad k=1,...N-1, \tag{C.23}$$

while the upwind stabilised one given by (C.22) does to

$$u\frac{\hat{q}_k-\hat{q}_{k-1}}{h}=0 \qquad k=1,...N-1. \tag{C.24}$$

For constant background flow u, the resulting central difference scheme solution $\hat{q}_{k+1} = \hat{q}_{k-1}$ admits alternating values and therefore spurious oscillatory solutions, while the upwind scheme $\hat{q}_k = \hat{q}_{k-1}$ leads to the expected constant solution. As a drawback, this type of advection discretisation suffers from a loss of accuracy in the order of h when compared to central differences, whose local discretisation error is in the order of h^2. In practise, this manifests itself in an overly diffusive effect, which, from a finite element point of view, can be remedied by increasing the order of the underlying finite element space.

Next to the analogy with the finite difference method, the classical DG upwind stabilisation method can also be seen from a finite volume point of view. For lowest order elements, the test functions ϕ are given by constant functions with compact support equal to a given cell K of the mesh. The upwind formulation (1.128) then reduces to

$$\int_K \rho_t \, dx = -\int_{\partial K} \mathbf{u}\cdot\mathbf{n}_K\tilde{\rho} \, dS \qquad \forall K \in \mathcal{T}, \tag{C.25}$$

for cell outward unit normal $\mathbf{n}_K$. This is analogous to a finite volume formulation, in the sense that the change of mass in a given cell is equal to a specified flux at the cell boundary. Similarly to the choice of polynomial order for the underlying finite element space, the finite volume scheme's diffusivity can be controlled by the polynomial order of the stencil used to generate the flux.

C.3 SUPG method

When presenting the SUPG method in Section 1.3.3, we noted that a stabilising effect follows from diffusion acting along streamlines (see (1.138)). The choice of diffusion in the direction of the flow can be motivated in the sense that orthogonally to the flow, the advection term vanishes, and therefore does not pose as a source of oscillations. Another argument for this choice can be made from the point of view of sub-grid discretisations, where the attribute 'sub-grid' refers to features that are smaller than the mesh grid size. In this section, we will further explore this approach.

One way to capture these small features and establish a connection between sub-grid scales and the SUPG method is by augmenting the finite element space to contain basis functions whose compact support is given by the interior of single elements. The unknowns associated with these additional functions are then solved for, thus contributing to an improved solution that contains information from within elements. This approach can give insight into possible choices of stabilisation parameter τ, and here we briefly illustrate this by means of the advection-diffusion equation, following [Bre00, ESW14]. For this purpose, we consider the weak form (C.4) with homogeneous boundary conditions, and will include a range of simplifying assumptions along the derivation. As mentioned above, the solution space V_{E_0} is augmented by a space B_{E_0} of bubble functions to

$$\hat{V}_{E_0} = V_{E_0} \oplus B_{E_0}, \tag{C.26}$$

so that we obtain a weak form

$$a(\hat{q}_h, \hat{p}_h) = \langle f, \hat{p}_h \rangle \qquad \forall \hat{p}_h \in \hat{V}_{E_0}, \tag{C.27}$$

for solution $\hat{q}_h \in \hat{V}_{E_0}$. Picking a test function of the form $c_h \in B_{E_0}$ with compact support in the interior of a given cell K, and using the decomposition $\hat{q}_h = q_h + b_h$ (for $b_h \in B_{E_0}$), we can reformulate the above to

$$a(b_h, c_h) = \langle f, c_h \rangle - a(q_h, c_h) = \int_K (f + \epsilon \Delta q_h - \mathbf{u} \cdot \nabla q_h) c_h \, dx, \tag{C.28}$$

where we have used integration by parts together with the compact support of c_h. By the same argument, we further find

$$a(b_h, c_h) = \int_K (-\epsilon \Delta b_h + \mathbf{u} \cdot \nabla b_h) c_h \, dx, \tag{C.29}$$

and assuming a sufficiently rich space B_{E_0}, we obtain a point-wise version of (C.28) within K, given by

$$-\epsilon\Delta b_h + \mathbf{u}\cdot\nabla b_h = q_{Kres}, \tag{C.30}$$

for cell-wise residual

$$q_{Kres} = \big(f + \epsilon\Delta q_h - \mathbf{u}\cdot\nabla q_h\big)|_K. \tag{C.31}$$

Assuming that the latter is constant, and writing the restriction of b_h to the cell K as $b_h|_K = d_K q_{Kres}$ (for unknown d_K to be determined), we obtain a differential equation for the interior of K given by

$$-\epsilon\Delta d_K + \mathbf{u}\cdot\nabla d_K = 1, \tag{C.32}$$

with $d_K|_{\partial K} = 0$.

Additionally, setting $\hat{p}_h = p_h \in V_{E_0}$ in (C.27), we obtain

$$a(q_h, p_h) = \langle f, p_h\rangle - a(b_h, p_h), \tag{C.33}$$

which is equal to the original weak form together with a new contribution in terms of b_h. Assuming piece-wise linear test functions p_h and a piece-wise constant advecting velocity, this contribution can be reformulated as

$$a(b_h, p_h) = \sum_K \int_\Omega (\epsilon\nabla b_h\cdot\nabla p_h + p_h\mathbf{u}\cdot\nabla b_h)\,dx = -\sum_K \int_K b_h(\mathbf{u}\cdot\nabla p_h)\,dx \tag{C.34}$$

$$= -\sum_K \int_K d_K q_{Kres}(\mathbf{u}\cdot\nabla p_h)\,dx = -\sum_K q_{Kres}(\mathbf{u}\cdot\nabla p_h)\int_K d_K\,dx \tag{C.35}$$

$$= -\sum_K \frac{1}{|K|}\int_K \big(q_{Kres}(\mathbf{u}\cdot\nabla p_h)\big)dx \int_K d_K\,dx \tag{C.36}$$

$$= -\langle q_{res}, \tau(\mathbf{u}\cdot\nabla p_h)\rangle, \tag{C.37}$$

where we have used integration by parts and $\Delta p_h = 0$, $\nabla\cdot\mathbf{u} = 0$ in the first line, and that q_{Kres} and $\mathbf{u}\cdot\nabla p_h$ are constant in the second. Further, we introduced a global residual q_{res} that is defined to be equal to the cell restricted one q_{Kres} for each cell K, and a parameter

τ such that its restriction $\tau|_K$ to a given cell K is defined by

$$\tau|_K = \frac{1}{|K|} \int_K d_K \, dx. \tag{C.38}$$

Substituting into (C.33), we arrive at

$$a(q_h, p_h) = \langle f, p_h \rangle + \langle q_{res}, \tau(\mathbf{u} \cdot \nabla p_h) \rangle \qquad \forall p_h \in V_{E_0}, \tag{C.39}$$

which corresponds to SUPG for the advection-diffusion equation. To see this, note that q_{res} corresponds to the strong formulation, implying that we obtain a streamline upwind contribution $\|\sqrt{\tau}\mathbf{u} \cdot \nabla p_h\|_2^2$ without loosing consistency. Consistency follows since if we apply the strong solution to (C.39), we have $q_{res} = 0$ and the stabilisation term disappears.

Next to bubble functions, there are a range of other approaches based on capturing information on unresolved scales, and the resulting stabilisation methods are known as variational multiscale stabilisation (VMS) [MKM+16]. For our purposes, there are two useful remarks that follow from the above derivation. First, the resulting residual based SUPG setup of the form (C.39) may be easier to formulate than a setup where all test functions are modified to include an upwind contribution (as in (1.136)). This can occur e.g. if the equation depends on other equations, so that a modification of the former requires a modification of the latter (see Chapter 4). Second, the derivation indicates that the stabilisation parameter τ is related to the solutions d_K from the interior of cells K. If the resolution is well-resolved, we can expect d_K, and therefore τ, to be small. This suggests the construction of residual based forms of τ, where the latter is a function of the magnitude of q_{res}. Aside from the SUPG context, such residual based formulations exist e.g. for sub-grid models, such as the one from [MNG15] considered in Remark 25 in Chapter 3.

C.3.1 Choice of stabilisation parameter

With the above in mind, we conclude the SUPG section with a discussion on possible choices for the stabilisation parameter τ. Specifically, we will first consider a τ parameter commonly used in the literature as well as variations thereof, followed by residual based τ parameters.

The former is based on analyses of simplified PDE scenarios via e.g. Green's functions

and Fourier analysis [MKM+16], and is given by

$$\tau = \frac{h}{2|\mathbf{u}|}. \tag{C.40}$$

Note that as $|\mathbf{u}| \to 0$, this value grows large. In the numerical implementation underlying the results presented in the main chapters above, this choice of τ causes computational challenges possibly related to round-off errors that negatively affect products of small and large numbers. To avoid this, the value can be further simplified by combining it with the so-called CFL number stability criterion, which commonly appears in the stability analysis of numerical schemes for advection-dominated flow [QSS10]:

$$\frac{\Delta t|\mathbf{u}|}{h} \leq 1. \tag{C.41}$$

Substituting (C.40) into the criterion, we find $\tau \geq \Delta t/2$, suggesting a minimum uniform stabilisation parameter of the form*

$$\tau = \frac{\Delta t}{2}, \tag{C.42}$$

which is used in Chapter 3. To arrive at a term closer to the standard one given by (C.40), we can combine the latter with (C.42) to arrive at

$$\tau = \left(\lambda \frac{2}{\Delta t} + \frac{2|\mathbf{u}|}{h}\right)^{-1}, \tag{C.43}$$

where $\lambda \geq 0$ is a tuning parameter. While the choice of τ according to (C.40) is well-known in the literature, to the best of my knowledge its definition according to (C.43) has been considered here for the first time. Using the CFL criterion, we find that the stabilisation parameter lies in the range

$$\tau \in \left[\frac{1}{\lambda+1}, \frac{1}{\lambda}\right] \cdot \frac{\Delta t}{2}, \tag{C.44}$$

where the upper limit is attained at $|\mathbf{u}| = 0$, while the lower one follows for velocity values reaching the limit $|\mathbf{u}| = h/\Delta t$ induced by the CFL criterion. Note that as $\lambda \to 0$, we recover the standard parameter (C.40), while at $\lambda = 1$, we obtain

$$\tau \in \left[\frac{\Delta t}{4}, \frac{\Delta t}{2}\right], \tag{C.45}$$

* The choice $\tau = \Delta t/2$ can also be justified via an argument involving the so-called Taylor-Galerkin method, which includes a second order approximation to the time derivative; for details see [Kuz10].

which corresponds to a small amount of diffusion, whose upper bound is given by the minimal value (C.42) derived using the CFL criterion. With this in mind, $\lambda \in [0,1]$ can be seen as a control parameter, allowing for more dissipation than $\tau = \Delta t/2$, while avoiding the numerical instability arising from $\tau = h/(2|\mathbf{u}|)$.

It is also possible to construct residual based τ parameters, and to illustrate this, we consider for simplicity a discretisation of the thermal field transport equation given by

$$\left\langle \gamma, \theta^a - \theta^n + \Delta t \frac{\mathbf{u}^{n+1,k} + \mathbf{u}^n}{2} \cdot \nabla \frac{\theta^a - \theta^n}{2} \right\rangle = 0 \qquad \forall \gamma \in \mathbb{V}_u, \tag{C.46}$$

noting that the setup discussed here can also be applied to more complex discretised equations, such as the ones considered in Chapters 3 and 4. The time-related superscripts are defined as in the numerical results Section 2.2 in Chapter 2, i.e. n denotes the current time level, $n+1, k$ the current guess for the next time level within the Picard iteration scheme, and a the advected quantity to be solved for. θ^a is then used to obtain the next guess $n+1, k+1$, and eventually the final value θ^{n+1}. Since this procedure does not solve exactly for the overall nonlinear system of equations derived from the discretisation, we obtain a non-zero residual of the form

$$\langle \gamma, \theta_{res} \rangle = \left\langle \gamma, \theta^{n+1} - \theta^n + \Delta t \frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{2} \cdot \nabla \frac{\theta^{n+1} - \theta^n}{2} \right\rangle \qquad \forall \gamma \in \mathbb{V}_u, \tag{C.47}$$

which we can use to define τ in analogy to the unresolved scales based definition (C.38) above. In view of the discussion above, which suggests a minimum value of $\tau = \Delta t/2$ and computational challenges for large values of τ, we find that potential residual based definitions are given by

$$\tau = \begin{cases} \dfrac{\Delta t}{2} & |\theta_{res}| < \alpha_0, \\ \dfrac{\Delta t}{2} \left(\ln \left(\dfrac{|\theta_{res}|}{\alpha_0} \right) + 1 \right) & \text{otherwise}, \end{cases} \tag{C.48}$$

and

$$\tau = \begin{cases} \dfrac{\Delta t}{2} & |\theta_{res}| < \alpha_0, \\ \dfrac{\Delta t}{2} \left(\dfrac{\beta - 1}{\alpha_1 - \alpha_0} \left(|\theta_{res}| - \alpha_0 \right) + 1 \right) & \alpha_0 \leq |\theta_{res}| \leq \alpha_1, \\ \dfrac{\Delta t}{2} \beta & \alpha_1 < |\theta_{res}|, \end{cases} \tag{C.49}$$

where $(\alpha_0, \alpha_1, \beta)$ is a set of threshold parameters that have to be tuned. Note that (C.48) corresponds to a log profile starting at $\Delta t/2$, while (C.49) corresponds to a linear increase starting at $\Delta t/2$ and stopping at a specified upper bound. As with the choice of τ according to (C.43), to the best of my knowledge its definition according to (C.48) and (C.49) has been described here for the first time.

In practise, we found in particular the linear profile to work well for a suitable choice of parameters $(\alpha_0, \alpha_1, \beta)$. However, the choice of tuning parameters is non-trivial, limiting the practicality of this approach; in future work, we aim to derive a systematic way of determining these parameters. For this reason, the numerical results in Chapters 3 and 4 are instead based on stabilisation parameters as given by the residual-free approach described above. Note that the residual-free approach is also more computationally efficient, in the sense that we are not required to compute θ_{res} at each time step (and similarly q_{res} for the SUPG method applied to the vorticity equation).

D

3D VORTICITY FORMULATION WITH BOUNDARIES

In this appendix, we extend the derivation of a vorticity equation in the presence of boundaries, as given in [BC18], to the three-dimensional thermal case. For this purpose, we consider a Poisson bracket of the form

$$\begin{aligned}\{F,H\} \coloneqq -\left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, \frac{\delta F}{\delta \mathbf{u}}\right) + G^{\theta}_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \frac{\delta F}{\delta \mathbf{u}}\right) \\ - G^{\rho}_{\mathbf{z}}\left(\frac{\delta F}{\delta \rho}, \frac{\delta H}{\delta \mathbf{u}}\right) - G^{\theta}_{\mathbf{z}}\left(\frac{\delta F}{\delta \theta}, \frac{\delta H}{\delta \mathbf{u}}\right),\end{aligned} \tag{D.1}$$

where as in Chapter 4, $\boldsymbol{\omega}$ denotes absolute vorticity. Further, $G^{\rho}_{\mathbf{z}}$ and $G^{\theta}_{\mathbf{z}}$ consist of bracket related terms that correspond to velocity forcing terms as well as advection terms for ρ and θ. Here, $\mathbf{z}$ denotes the state $(\rho, \mathbf{u}, \theta)$, noting that $G^{\rho}_{\mathbf{z}}$ and $G^{\theta}_{\mathbf{z}}$ may contain terms involving these fields. For instance, for the standard bracket given in Definition 5, they are given by

$$G^{\rho}_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, \frac{\delta F}{\delta \mathbf{u}}\right) = \left\langle \frac{\delta H}{\delta \rho}, \nabla \cdot \frac{\delta F}{\delta \mathbf{u}} \right\rangle, \qquad G^{\theta}_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \frac{\delta F}{\delta \mathbf{u}}\right) = \left\langle \frac{1}{\rho}\frac{\delta H}{\delta \theta}\nabla\theta, \frac{\delta F}{\delta \mathbf{u}} \right\rangle, \tag{D.2}$$

while for the temperature SUPG almost Poisson bracket in Definition 19, the thermal contribution reads

$$G^{\theta}_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \frac{\delta F}{\delta \mathbf{u}}\right) = L^{\theta}_{\frac{\delta F}{\delta \mathbf{u}}/\rho}\left(\theta; \left(\frac{\delta H}{\delta \theta}\right)_{\mathbf{u}}\right) + \left\langle \left(\widetilde{\widetilde{\frac{\delta H}{\delta \theta}}}\right)_{\mathbf{u}}, \mathbb{S}^{-1}_{\mathbf{u}}\left(\theta^{a}\left(\frac{\delta F}{\delta \mathbf{u}}\right)\right)\right\rangle . \tag{D.3}$$

Using this notation, we obtain a momentum equation of the form

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = -\left\langle \mathbf{w}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, \mathbf{w}\right) + G^{\theta}_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \mathbf{w}\right) \qquad \forall \mathbf{w} \in \mathbb{V}_u, \tag{D.4}$$

where $\mathbb{V}_u$ denotes the discrete velocity space. To further obtain a vorticity evolution equation, we consider the weak diagnostic vorticity equation, which in the absence of boundaries is given by

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega} \rangle = \langle \nabla \times \boldsymbol{\Sigma}, \mathbf{u} \rangle + \langle \boldsymbol{\Sigma}, 2\boldsymbol{\Omega} \rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_{\omega}, \tag{D.5}$$

where $\mathbb{V}_{\omega}$ denotes the discrete vorticity space. Differentiating the diagnostic vorticity equation with respect to time, we arrive at

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle = \langle \nabla \times \boldsymbol{\Sigma}, \mathbf{u}_t \rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_{\omega}. \tag{D.6}$$

Since in the compatible finite element context, $\nabla \times \boldsymbol{\Sigma} \in \mathbb{V}_u$ for any $\boldsymbol{\Sigma} \in \mathbb{V}_{\omega}$, we may then substitute the right-hand side term in (D.6) by the right-hand side of the momentum equation (D.4). The resulting vorticity evolution equation is then given by

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle = -\left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma}\right) + G^{\theta}_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma}\right) \quad \forall \boldsymbol{\Sigma} \in \mathbb{V}_{\omega}. \tag{D.7}$$

In the presence of boundaries, we instead consider a velocity space given by

$$\mathring{\mathbb{V}}_u = \{\mathbf{v} \in \mathbb{V}_u \mid (\mathbf{v} \cdot \mathbf{n})|_{\partial\Omega} = 0\}, \tag{D.8}$$

where $|_{\partial\Omega}$ denotes the trace operator on the boundary of Ω. In this case, the curl of functions $\boldsymbol{\Sigma} \in \mathbb{V}_{\omega}$ need not be in the velocity space $\mathring{\mathbb{V}}_u$ any longer. To circumnavigate this problem, the vorticity space is split into an interior and an exterior part [BC18], i.e.

$$\mathring{\mathbb{V}}_{\omega} = \{\boldsymbol{\Sigma} \in \mathbb{V}_{\omega} \mid (\mathbf{n} \times \boldsymbol{\Sigma})|_{\partial\Omega} = 0\}, \tag{D.9}$$

$$\mathbb{V}^{\perp}_{\omega}: \quad \mathbb{V}_{\omega} = \mathring{\mathbb{V}}_{\omega} \oplus \mathbb{V}^{\perp}_{\omega}, \tag{D.10}$$

that is $\mathbb{V}_\omega^\perp$ is defined as the L^2-orthogonal complement of the interior space $\mathring{\mathbb{V}}_\omega$ in $\mathbb{V}_\omega$. The vorticity is then split into an exterior part $\mathbf{Z}'$ and an interior part $\mathring{\mathbf{Z}}$, where the latter is defined diagnostically, while the former is considered as a prognostic variable that is part of the Poisson bracket framework. The proposition below is an extension of the framework presented in [BC18] to the three-dimensional thermal case, for general terms G corresponding to the bracket's density and thermal type field part.

Proposition 39. *Consider a discretised almost Poisson bracket defined by*

$$\begin{aligned}\{F,H\} \coloneqq \quad & G^\rho_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, D_{\mathbf{u},\mathbf{Z}'}(F)\right) + G^\theta_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, D_{\mathbf{u},\mathbf{Z}'}(F)\right) - \left\langle \nabla \times \frac{\delta F}{\delta \mathbf{Z}'}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle \\ & - G^\rho_{\mathbf{z}}\left(\frac{\delta F}{\delta \rho}, D_{\mathbf{u},\mathbf{Z}'}(H)\right) - G^\theta_{\mathbf{z}}\left(\frac{\delta F}{\delta \theta}, D_{\mathbf{u},\mathbf{Z}'}(H)\right) + \left\langle \nabla \times \frac{\delta H}{\delta \mathbf{Z}'}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta F}{\delta \mathbf{u}}\right\rangle \\ & - \left\langle \frac{\delta F}{\delta \mathbf{u}}, \boldsymbol{\omega} \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle,\end{aligned} \tag{D.11}$$

for $G^\rho_{\mathbf{z}}$ and $G^\theta_{\mathbf{z}}$ defined as described above, and $D_{\mathbf{u},\mathbf{Z}'}$ defined on functionals K such that

$$D_{\mathbf{u},\mathbf{Z}'}(K) = \frac{\delta K}{\delta \mathbf{u}} + \nabla \times \frac{\delta K}{\delta \mathbf{Z}'}. \tag{D.12}$$

Further, assume the diagnostic vorticity $\boldsymbol{\omega}(\mathbf{u}, \mathbf{Z}') \in \mathbb{V}_\omega$ to be defined by

$$\langle \mathring{\boldsymbol{\Sigma}}, \mathring{\mathbf{Z}}\rangle = \langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \mathbf{u}\rangle + \langle \mathring{\boldsymbol{\Sigma}}, 2\boldsymbol{\Omega}\rangle \qquad \forall \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_\omega, \tag{D.13}$$

$$\left\langle \boldsymbol{\Sigma}, \boldsymbol{\omega} - \left(\mathring{\mathbf{Z}} + \mathbf{Z}'\right)\right\rangle = 0 \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega, \tag{D.14}$$

for $\mathring{\mathbf{Z}} \in \mathring{\mathbb{V}}_\omega$. Then for the shallow water, thermal shallow water, and compressible Euler Hamiltonians introduced in Section 1.2, as well as the MHD Hamiltonian given in Remark 32, the Poisson system (1.53) *together with this bracket admits an energy conserving system of equations given by*

$$\langle \mathbf{w}, \mathbf{u}_t\rangle = -\left\langle \mathbf{w}, \boldsymbol{\omega}(\mathbf{u}, \mathbf{Z}') \times \frac{1}{\rho}\frac{\delta H}{\delta \mathbf{u}}\right\rangle + G^\rho_{\mathbf{z}}\left(\frac{\delta H}{\delta \rho}, \mathbf{w}\right) + G^\theta_{\mathbf{z}}\left(\frac{\delta H}{\delta \theta}, \mathbf{w}\right) \qquad \forall \mathbf{w} \in \mathring{\mathbb{V}}_u, \tag{D.15}$$

$$\langle \boldsymbol{\Sigma}', \mathbf{Z}'_t\rangle = -\left\langle \nabla \times \boldsymbol{\Sigma}', \boldsymbol{\omega}(\mathbf{u}, \mathbf{Z}') \times \tfrac{1}{\rho}\tfrac{\delta H}{\delta \mathbf{u}}\right\rangle + G^\rho_{\mathbf{z}}\left(\tfrac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma}'\right) + G^\theta_{\mathbf{z}}\left(\tfrac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma}'\right) \quad \forall \boldsymbol{\Sigma}' \in \mathbb{V}_\omega^\perp, \tag{D.16}$$

$$\langle \phi, \rho_t\rangle = -G^\rho_{\mathbf{z}}\left(\phi, \frac{\delta H}{\delta \mathbf{u}}\right) \qquad \forall \phi \in \mathbb{V}_\rho, \tag{D.17}$$

$$\langle \gamma, \theta_t\rangle = -G^\theta_{\mathbf{z}}\left(\gamma, \frac{\delta H}{\delta \mathbf{u}}\right) \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{D.18}$$

Further, assume that $\boldsymbol{\omega}$ *fulfils initial conditions of the form*

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_0 \rangle = \langle \nabla \times \boldsymbol{\Sigma}, \mathbf{u}_0 \rangle + \langle\!\langle \boldsymbol{\Sigma}, (\mathbf{n} \times \mathbf{u}_0)|_{\partial\Omega} \rangle\!\rangle + \langle \boldsymbol{\Sigma}, 2\boldsymbol{\Omega} \rangle \qquad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega, \tag{D.19}$$

for initial values $\boldsymbol{\omega}_0 = \boldsymbol{\omega}(\mathbf{x}, 0)$ *as well as* $\mathbf{u}_0 = \mathbf{u}(\mathbf{x}, 0)$*, and* $L^2(\partial\Omega)$*-inner product* $\langle\!\langle ., . \rangle\!\rangle$*. Then the system* (D.15) - (D.18) *is equivalent to*

$$\langle \mathbf{w}, \mathbf{u}_t \rangle = -\left\langle \mathbf{w}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^\rho_{\mathbf{z}} \left(\frac{\delta H}{\delta \rho}, \mathbf{w} \right) + G^\theta_{\mathbf{z}} \left(\frac{\delta H}{\delta \theta}, \mathbf{w} \right) \qquad \forall \mathbf{w} \in \mathring{\mathbb{V}}_u, \tag{D.20}$$

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle = -\left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \tfrac{1}{\rho} \tfrac{\delta H}{\delta \mathbf{u}} \right\rangle + G^\rho_{\mathbf{z}} \left(\tfrac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma} \right) + G^\theta_{\mathbf{z}} \left(\tfrac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma} \right) \quad \forall \boldsymbol{\Sigma} \in \mathbb{V}_\omega, \tag{D.21}$$

$$\langle \phi, \rho_t \rangle = -G^\rho_{\mathbf{z}} \left(\phi, \frac{\delta H}{\delta \mathbf{u}} \right) \qquad \forall \phi \in \mathbb{V}_\rho, \tag{D.22}$$

$$\langle \gamma, \theta_t \rangle = -G^\theta_{\mathbf{z}} \left(\gamma, \frac{\delta H}{\delta \mathbf{u}} \right) \qquad \forall \gamma \in \mathbb{V}_\theta. \tag{D.23}$$

Proof. *Since none of the specified Hamiltonians contain the boundary vorticity field* $\mathbf{Z}'$*, all appearances of* $\frac{\delta H}{\delta \mathbf{Z}'}$ *in the bracket are equal to zero. Further, all appearances of* $\frac{\delta F}{\delta \mathbf{Z}'}$ *are also equal to zero for choices of functionals* F *which do not contain* $\mathbf{Z}'$*. In particular, this is the case for* $F = \langle \mathbf{w}, \mathbf{u} \rangle$*,* $F = \langle \phi, \rho \rangle$*, and* $F = \langle \gamma, \theta \rangle$*, which are the functionals used in the Poisson system to obtain the momentum, density, and thermal field equations. Ignoring these appearances reduces bracket* (D.11) *to the usual one given by* (D.1)*, since in this case the last terms of the first two lines in* (D.11) *vanish, and* $D_{\mathbf{u},\mathbf{Z}'}$ *reduces to* $D_{\mathbf{u}}(K) = \frac{\delta K}{\delta \mathbf{u}}$*. Using this observation, equations* (D.15)*,* (D.17)*, and* (D.18) *follow readily. To obtain the evolution equation for* $\mathbf{Z}'$*, we pick a functional of the form* $F = \langle \boldsymbol{\Sigma}', \mathbf{Z}' \rangle$*, for any* $\boldsymbol{\Sigma}' \in \mathbb{V}_\omega^\perp$*. In this case, only the first line of* (D.11) *contains non-zero terms. For this choice of functional, we have* $\frac{\delta F}{\delta \mathbf{Z}'} = \mathbf{Z}'$ *and further*

$$D_{\mathbf{u},\mathbf{Z}'}(F) = \nabla \times \boldsymbol{\Sigma}', \tag{D.24}$$

which then yields the required evolution equation (D.16)*. Finally, we note that bracket* (D.11) *is indeed an almost Poisson bracket, since it is clearly bilinear, and antisymmetry follows since the terms in the first and second lines of* (D.11) *form antisymmetric pairs, and the velocity advection term in the third line is antisymmetric in itself.*

To show that (D.15) - (D.18) *implies the formulation* (D.20) - (D.23)*, we consider the*

defining equations (D.13) *and* (D.14) *for* $\boldsymbol{\omega}$ *to arrive at*

$$\langle \boldsymbol{\Sigma}, \boldsymbol{\omega}_t \rangle = \left\langle \boldsymbol{\Sigma}, \left(\mathring{\mathbf{Z}} + \mathbf{Z}'\right)_t \right\rangle = \left\langle \mathring{\boldsymbol{\Sigma}} + \boldsymbol{\Sigma}', \left(\mathring{\mathbf{Z}} + \mathbf{Z}'\right)_t \right\rangle = \left\langle \mathring{\boldsymbol{\Sigma}}, \mathring{\mathbf{Z}}_t \right\rangle + \langle \boldsymbol{\Sigma}', \mathbf{Z}'_t \rangle \tag{D.25}$$

$$= \langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \mathbf{u}_t \rangle - \left\langle \nabla \times \boldsymbol{\Sigma}', \boldsymbol{\omega} \times \tfrac{1}{\rho} \tfrac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}} \left(\tfrac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma}' \right) + G^{\theta}_{\mathbf{z}} \left(\tfrac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma}' \right) \tag{D.26}$$

$$= - \left\langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}} \left(\frac{\delta H}{\delta \rho}, \nabla \times \mathring{\boldsymbol{\Sigma}} \right) + G^{\theta}_{\mathbf{z}} \left(\frac{\delta H}{\delta \theta}, \nabla \times \mathring{\boldsymbol{\Sigma}} \right) \tag{D.27}$$

$$- \left\langle \nabla \times \boldsymbol{\Sigma}', \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}} \left(\frac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma}' \right) + G^{\theta}_{\mathbf{z}} \left(\frac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma}' \right) \tag{D.28}$$

$$= - \left\langle \nabla \times \boldsymbol{\Sigma}, \boldsymbol{\omega} \times \frac{1}{\rho} \frac{\delta H}{\delta \mathbf{u}} \right\rangle + G^{\rho}_{\mathbf{z}} \left(\frac{\delta H}{\delta \rho}, \nabla \times \boldsymbol{\Sigma} \right) + G^{\theta}_{\mathbf{z}} \left(\frac{\delta H}{\delta \theta}, \nabla \times \boldsymbol{\Sigma} \right), \tag{D.29}$$

for all functions $\boldsymbol{\Sigma} \in \mathbb{V}_{\omega}$. *Note that we used the orthogonal decomposition* $\boldsymbol{\Sigma} = \mathring{\boldsymbol{\Sigma}} + \boldsymbol{\Sigma}'$, *for* $\mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{\omega}$ *and* $\boldsymbol{\Sigma}' \in \mathbb{V}'_{\omega}$. *Further we used the momentum equation* (D.15), *noting that for interior vorticity functions* $\mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{\omega}$, *we have* $\nabla \times \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{u}$. *Finally, in the last line we made use of the bilinearity of* G.

For the converse, we define $\mathring{\mathbf{Z}}$ *and* $\mathbf{Z}'$ *via* $\boldsymbol{\omega}$ *as their projections into the interior and boundary vorticity spaces, respectively:*

$$\mathring{\mathbf{Z}} = P_{\mathring{\mathbb{V}}_{\omega}}(\boldsymbol{\omega}), \qquad \mathbf{Z}' = P_{\mathbb{V}^{\perp}_{\omega}}(\boldsymbol{\omega}). \tag{D.30}$$

The diagnostic equation (D.14) *then follows immediately using the orthogonal decomposition, and so does the evolution equation* (D.16), *for which we set* $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}'$ *in* (D.21). *It remains to derive the second diagnostic equation* (D.13). *The latter holds initially given the initialisation* (D.19) *with* $\boldsymbol{\Sigma}$ *set to* $\mathring{\boldsymbol{\Sigma}}$, *noting that in this case the boundary term vanishes. To demonstrate that it holds for all time, we differentiate it with respect to time and arrive at*

$$\frac{d}{dt} \left(\langle \boldsymbol{\Sigma}, \mathring{\mathbf{Z}} \rangle - \langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \mathbf{u} \rangle - \langle \mathring{\boldsymbol{\Sigma}}, 2\boldsymbol{\Omega} \rangle \right) = 0 \qquad \forall \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{\omega}, \tag{D.31}$$

$$\iff \langle \mathring{\boldsymbol{\Sigma}}, \mathring{\mathbf{Z}}_t \rangle = \langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \mathbf{u}_t \rangle \qquad \forall \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{\omega}, \tag{D.32}$$

$$\iff \langle \mathring{\boldsymbol{\Sigma}}, \boldsymbol{\omega}_t \rangle = \langle \nabla \times \mathring{\boldsymbol{\Sigma}}, \mathbf{u}_t \rangle \qquad \forall \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{\omega}. \tag{D.33}$$

Finally, considering the momentum and vorticity equations (D.20) *and* (D.21), *respectively, and noting that* $\nabla \times \mathring{\boldsymbol{\Sigma}} \in \mathring{\mathbb{V}}_{u}$, *we find that the left- and right-hand sides above are*

indeed equal. □

Note that while all terms in the almost Poisson bracket (D.11) containing $\frac{\delta H}{\delta \mathbf{Z}'}$ play no role in the resulting prognostic equations, they are still included to ensure that the bracket is indeed antisymmetric. Further, note that the proof of Proposition 39 works equally if all occurrences of $\boldsymbol{\omega}$ *in the bracket* are replaced by the SUPG modified form $\boldsymbol{\omega} - \tau \boldsymbol{\omega}_{res}$. Finally, the equivalence between the $\mathbf{Z}'$ and $\boldsymbol{\omega}$ based setups ensures that we can use the prognostic equations (D.20) - (D.23) while retaining energy conservation, since its equivalent $\mathbf{Z}'$ based form is derived from an almost Poisson bracket.

References

[ABCM02] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis*, 39(5):1749–1779, 2002.

[ABF05] D. N. Arnold, D. Boffi, and R. S. Falk. Quadrilateral H (div) finite elements. *SIAM Journal on Numerical Analysis*, 42(6):2429–2451, 2005.

[Abg06] R. Abgrall. Residual distribution schemes: current status and future trends. *Computers & Fluids*, 35(7):641–669, 2006.

[AFW06] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.

[AFW10] D. N. Arnold, R. S. Falk, and R. Winther. Finite element exterior calculus: from Hodge theory to numerical stability. *Bulletin of the American mathematical society*, 47(2):281–354, 2010.

[AK99] V. I. Arnold and B. A. Khesin. *Topological methods in hydrodynamics*, volume 125. Springer Science & Business Media, 1999.

[AL77] A. Arakawa and V. R. Lamb. Computational design of the basic dynamical processes of the UCLA general circulation model. *General circulation models of the atmosphere*, 17(Supplement C):173–265, 1977.

[Arn18] D. N. Arnold. *Finite element exterior calculus*. SIAM, 2018.

[BAA+19] S. Balay, S. Abhyankar, M. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, A. Dener, V. Eijkhout, W. D. Gropp, et al. PETSc users manual. 2019.

[BBF+13] D. Boffi, F. Brezzi, M. Fortin, et al. *Mixed finite element methods and applications*, volume 44. Springer, 2013.

[BC18] W. Bauer and C. J. Cotter. Energy–enstrophy conserving compatible finite element schemes for the rotating shallow water equations with slip boundary conditions. *Journal of Computational Physics*, 373:171 – 187, 2018.

[BDM85] F. Brezzi, J. Douglas, and L. D. Marini. Two families of mixed finite elements for second order elliptic problems. *Numerische Mathematik*, 47(2):217–235, 1985.

[BGB19] W. Bauer and F. Gay-Balmaz. Towards a geometric variational discretization of compressible fluids: the rotating shallow water equations. *Journal of Computational Dynamics*, 6(2158-2491):1, 2019.

[BGMS97] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object-oriented numerical software libraries. In *Modern software tools for scientific computing*, pages 163–202. Springer, 1997.

[BGS+19] 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. *Quarterly Journal of the Royal Meteorological Society*, 2019.

[BH82] A. N. Brooks and T. J. R. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 32(1-3):199–259, 1982.

[BH04] E. Burman and P. Hansbo. Edge stabilization for Galerkin approximations of convection–diffusion–reaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(15-16):1437–1453, 2004.

[BKN15] L. Bao, R. Klöfkorn, and R. D. Nair. Horizontally explicit and vertically implicit (HEVI) time discretization scheme for a discontinuous Galerkin nonhydrostatic model. *Monthly Weather Review*, 143(3):972–990, 2015.

[BMH+16] G.-T. Bercea, A. T. T. McRae, D. A. Ham, L. Mitchell, F. Rathgeber, L. Nardi, F. Luporini, and P. H. J. Kelly. A structure-exploiting numbering algorithm for finite elements on extruded meshes, and its performance evaluation in Firedrake. *Geoscientific Model Development*, 9(10):3803–3815, 2016.

[BMR98] F. Brezzi, D. Marini, and A. Russo. Applications of the pseudo residual-free bubbles to the stabilization of convection-diffusion problems. *Computer Methods in Applied Mechanics and Engineering*, 166(1-2):51–63, 1998.

[BO06] O. Bokhove and M. Oliver. Parcel Eulerian–Lagrangian fluid dynamics of rotating geophysical flows. *Proceedings of the Royal Society A: mathematical, physical and engineering sciences*, 462(2073):2575–2592, 2006.

[BR09] P. B. Bochev and D. Ridzal. Rehabilitation of the lowest-order Raviart–Thomas element on quadrilateral grids. *SIAM Journal on Numerical Analysis*, 47(1):487–507, 2009.

[Bre00] F. Brezzi. Interacting with the subgrid world. *Chapman and Hall/CRC*, pages 69–82, 2000.

[CH11] D. Cohen and E. Hairer. Linear energy-preserving integrators for Poisson systems. *BIT Numerical Mathematics*, 51(1):91–101, 2011.

[Cia13] P. G. Ciarlet. *Linear and nonlinear functional analysis with applications*, volume 130. SIAM, 2013.

[CJDWH90] R. L. Carpenter Jr, K. K. Droegemeier, P. R. Woodward, and C. E. Hane. Application of the piecewise parabolic method (PPM) to meteorological modeling. *Monthly Weather Review*, 118(3):586–612, 1990.

[CKS00] B. Cockburn, G. E. Karniadakis, and C. W. Shu. The development of discontinuous Galerkin methods. In *Discontinuous Galerkin Methods*, pages 3–50. Springer, 2000.

[CM14] C. J. Cotter and A. T. T. McRae. Compatible finite element methods for numerical weather prediction. *arXiv preprint arXiv:1401.0616*, 2014.

[CS12] C. J. Cotter and J. Shipton. Mixed finite elements for numerical weather prediction. *Journal of Computational Physics*, 231(21):7076 – 7091, 2012.

[DHLM05] M. Desbrun, A. N. Hirani, M. Leok, and J. E. Marsden. Discrete exterior calculus. *arXiv preprint math/0508341*, 2005.

[DR04] M. Dixon and S. Reich. Symplectic time-stepping for particle methods. *GAMM-Mitteilungen*, 27(1):9–24, 2004.

[EDK19] C. Eldred, T. Dubos, and E. Kritsikis. A quasi-Hamiltonian discretization of the thermal shallow water equations. *Journal of Computational Physics*, 379:1 – 31, 2019.

[ESW14] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, USA, 2014.

[FGH+13] R. Ford, M. J. Glover, D. A. Ham, C. M. Maynard, S. M. Pickles, G. Riley, and N. Wood. Gung Ho: A code design for weather and climate prediction on exascale machines. In *Proceedings of the Exascale Applications and Software Conference*, 2013.

[Gas11] A. Gassmann. Inspection of hexagonal and triangular C-grid discretizations of the shallow water equations. *Journal of Computational Physics*, 230(7):2706–2721, 2011.

[Gas13] A. Gassmann. A global hexagonal C-grid non-hydrostatic dynamical core (ICON-IAP) designed for energetic consistency. *Quarterly Journal of the Royal Meteorological Society*, 139(670):152–175, 2013.

[GH01] D. Gottlieb and J. S. Hesthaven. Spectral methods for hyperbolic problems. *Journal of Computational and Applied Mathematics*, 128(1-2):83–131, 2001.

[GH08] A. Gassmann and H. J. Herzog. Towards a consistent numerical compressible non-hydrostatic model using generalized Hamiltonian tools. *Quarterly Journal of the Royal Meteorological Society*, 134(635):1597–1613, 2008.

[GHK+09] M. Giorgetta, T. Hundertmark, P. Korn, S. Reich, and M. Restelli. Conservative space and time regularizations for the ICON model. *Reports on Earth system science*, 67(1-32), 2009.

[GMHC20] 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. *Geoscientific Model Development*, 13(2):735–761, 2020.

[GSP04] J. Galewsky, R. K. Scott, and L. M. Polvani. An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus A: Dynamic Meteorology and Oceanography*, 56(5):429–440, 2004.

[HKRB83] A. Hollingsworth, P. Kållberg, V. Renner, and D. M. Burridge. An internal symmetric computational instability. *Quarterly Journal of the Royal Meteorological Society*, 109(460):417–428, 1983.

[HLW06] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[HMLH18] M. Homolya, L. Mitchell, F. Luporini, and D. A. Ham. TSFC: a structure-preserving form compiler. *SIAM Journal on Scientific Computing*, 40(3):C401–C428, 2018.

[HMR98] 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*, 137(1):1–81, 1998.

[JW11] C. Jablonowski and D. L. Williamson. The pros and cons of diffusion, filters and fixers in atmospheric general circulation models. In *Numerical techniques for global atmospheric models*, pages 381–493. Springer, 2011.

[KLT12] D. Kuzmin, R. Löhner, and S. Turek. *Flux-corrected transport: principles, algorithms, and applications*. Springer, 2012.

[KM18] R. C. Kirby and L. Mitchell. Solver composition across the PDE/linear algebra barrier. *SIAM Journal on Scientific Computing*, 40(1):C76–C98, 2018.

[KS06] R. M. Kirby and S. J. Sherwin. Stabilisation of spectral/hp element methods through spectral vanishing viscosity: Application to fluid mechanics modelling. *Computer Methods in Applied Mechanics and Engineering*, 195(23-24):3128–3144, 2006.

[Kuz10] D. Kuzmin. A guide to numerical methods for transport equations. *University Erlangen-Nuremberg*, 2010.

[LHK17] F. Luporini, D. A. Ham, and P. H. J. Kelly. An algorithm for the optimization of finite element integration loops. *ACM Transactions on Mathematical Software (TOMS)*, 44(1):1–26, 2017.

[LP18] D. Lee and A. Palha. A mixed mimetic spectral element model of the rotating shallow water equations on the cubed sphere. *Journal of Computational Physics*, 375:240–262, 2018.

[LP20] D. Lee and A. Palha. A mixed mimetic spectral element model of the 3D compressible Euler equations on the cubed sphere. *Journal of Computational Physics*, 401:108993, 2020.

[LR04] B. Leimkuhler and S. Reich. *Simulating hamiltonian dynamics*, volume 14. Cambridge University Press, 2004.

[LR11] V. Lucarini and F. Ragone. Energetics of climate models: net energy balance and meridional enthalpy transport. *Reviews of Geophysics*, 49(1), 2011.

[MBM+16] A. T. 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*, 38(5):S25–S47, 2016.

[MBTC18] T. Melvin, T. Benacchio, J. Thuburn, and C. J. Cotter. Choice of function spaces for thermodynamic variables in mixed finite-element methods. *Quarterly Journal of the Royal Meteorological Society*, 144(712):900–916, 2018.

[MC14] A. T. 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*, 140(684):2223–2234, 2014.

[MDW+10] T. Melvin, M. Dubal, N. Wood, A. Staniforth, and M. Zerroukat. An inherently mass-conserving iterative semi-implicit semi-Lagrangian discretization of the non-hydrostatic vertical-slice equations. *Quarterly Journal of the Royal Meteorological Society*, 136(648):799–814, 2010.

[MG80] P. J. Morrison and J. M. Greene. Noncanonical Hamiltonian density formulation of hydrodynamics and ideal magnetohydrodynamics. *Physical Review Letters*, 45(10):790, 1980.

[MKM+16] S. Marras, J. F. Kelly, M. Moragues, A. Müller, M. A. Kopera, M. Vázquez, F. X. Giraldo, G. Houzeaux, and O. Jorba. A review of element-based Galerkin methods for numerical weather prediction: finite elements, spectral elements, and discontinuous Galerkin. *Archives of Computational Methods in Engineering*, 23(4):673–722, 2016.

[MNG15] S. Marras, M. Nazarov, and F. X. Giraldo. Stabilized high-order Galerkin methods based on a parameter-free dynamic SGS model for LES. *Journal of Computational Physics*, 301:77–101, 2015.

[Mor82] P. J. Morrison. Poisson brackets for fluids and plasmas. In *AIP Conference proceedings*, volume 88, pages 13–46. American Institute of Physics, 1982.

[Mor98] P. J. Morrison. Hamiltonian description of the ideal fluid. *Reviews of Modern Physics*, 70(2):467, 1998.

[MR13] J. E. Marsden and T. S. Ratiu. *Introduction to mechanics and symmetry: a basic exposition of classical mechanical systems*, volume 17. Springer Science & Business Media, 2013.

[NC17a] A. Natale and C. J. Cotter. Scale-selective dissipation in energy-conserving finite element schemes for two-dimensional turbulence. *Quarterly Journal of the Royal Meteorological Society*, 143(705):1734–1745, 2017.

[NC17b] A. Natale and C. J. Cotter. A variational H(div) finite element discretisation for perfect incompressible fluids. *IMA Journal of Numerical Analysis*, 38(3):1388–1419, 2017.

[NSC16] A. Natale, J. Shipton, and C. J. Cotter. Compatible finite element spaces for geophysical fluid dynamics. *Dynamics and Statistics of the Climate System*, 1(1), 2016.

[OT79] S. A. Orszag and C.-M. Tang. Small-scale structure of two-dimensional magnetohydrodynamic turbulence. *Journal of Fluid Mechanics*, 90(1):129–143, 1979.

[QSS10] A. Quarteroni, R. Sacco, and F. Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.

[QV08] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.

[Rei99] S. Reich. Backward error analysis for numerical integrators. *SIAM Journal on Numerical Analysis*, 36(5):1549–1570, 1999.

[RHM+16] F. Rathgeber, D. A. Ham, L. Mitchell, M. Lange, F. Luporini, A. T. T. McRae, G. T. Bercea, G. R. Markall, and P. H. J. Kelly. Firedrake: automating the finite element method by composing abstractions. *ACM Transactions on Mathematical Software (TOMS)*, 43(3):1–27, 2016.

[Rip93] P Ripa. Conservation laws for primitive equations models with inhomogeneous layers. *Geophysical & Astrophysical Fluid Dynamics*, 70(1-4):85–111, 1993.

[RKL10] M. E. Rognes, R. C. Kirby, and A. Logg. Efficient assembly of H(div) and H(curl) conforming finite elements. *SIAM Journal on Scientific Computing*, 31(6):4130–4151, 2010.

[RTKS10] T. D. Ringler, J. Thuburn, J. B. Klemp, and W. C. Skamarock. A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids. *Journal of Computational Physics*, 229(9):3065–3090, 2010.

[Sal04] R. Salmon. Poisson-bracket approach to the construction of energy- and potential-enstrophy-conserving algorithms for the shallow-water equations. *Journal of the Atmospheric Sciences*, 61(16):2016–2036, 2004.

[SGC18] J. Shipton, T. H. Gibson, and C. J. Cotter. Higher-order compatible finite element schemes for the nonlinear rotating shallow water equations on the sphere. *Journal of Computational Physics*, 375:1121–1137, 2018.

[She90] T. G. Shepherd. Symmetries, conservation laws, and Hamiltonian structure in geophysical fluid dynamics. In *Advances in Geophysics*, volume 32, pages 287–338. Elsevier, 1990.

[SKD+12] W. C. Skamarock, J. B. Klemp, M. G. Duda, L. D. Fowler, S.-H. Park, and T. D. Ringler. A multiscale nonhydrostatic atmospheric model using centroidal Voronoi tesselations and C-grid staggering. *Monthly Weather Review*, 140(9):3090–3105, 2012.

[Sma63] J. Smagorinsky. General circulation experiments with the primitive equations: I. The basic experiment. *Monthly Weather Review*, 91(3):99–164, 1963.

[ST12] A. Staniforth and J. Thuburn. Horizontal grids for global weather and climate prediction models: a review. *Quarterly Journal of the Royal Meteorological Society*, 138(662):1–26, 2012.

[SWW+93] 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*, 17(1):1–22, 1993.

[TCD14] J. Thuburn, C. J. Cotter, and T. Dubos. A mimetic, semi-implicit, forward-in-time, finite volume shallow water model: comparison of hexagonal-icosahedral and cubed-sphere grids. *Geoscientific Model Development*, 7(3):909–929, 2014.

[TD16] P. Tsoutsanis and D. Drikakis. Addressing the challenges of implementation of high-order finite volume schemes for atmospheric dynamics of unstructured meshes. European Congress on Computational Methods in Applied Sciences and Engineering, 2016.

[TF10] M. A. Taylor and A. Fournier. A compatible and conservative spectral element method on unstructured grids. *Journal of Computational Physics*, 229(17):5879–5895, 2010.

[TGS+20] M. A. Taylor, O. Guba, A. Steyer, P. A. Ullrich, D. M. Hall, and C. Eldred. An energy consistent discretization of the nonhydrostatic equations in primitive variables. *Journal of Advances in Modeling Earth Systems*, 12(1), 2020.

[Thu08] John Thuburn. Some conservation issues for the dynamical cores of NWP and climate models. *Journal of Computational Physics*, 227(7):3715–3730, 2008.

[Tor01] E. F. Toro. Shock-capturing methods for free-surface shallow fluids. *Wiley, Chichester UK*, 2001.

[TRSK09] J. Thuburn, T. D. Ringler, W. C. Skamarock, and J. B. Klemp. Numerical representation of geostrophic modes on arbitrarily structured C-grids. *Journal of Computational Physics*, 228(22):8321–8335, 2009.

[Val17] G. K. Vallis. *Atmospheric and oceanic fluid dynamics*. Cambridge University Press, 2017.

[WCB20a] G. A. Wimmer, C. J. Cotter, and W. Bauer. Energy conserving SUPG methods for compatible finite element schemes in numerical weather prediction. *arXiv preprint arXiv:2001.09590*, 2020.

[WCB20b] G. A. Wimmer, C. J. Cotter, and W. Bauer. Energy conserving upwinded compatible finite element schemes for the rotating shallow water equations. *Journal of Computational Physics*, 401:109016, 2020.

[WDH+92] 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*, 102(1):211–224, 1992.

[WTC12] H. Weller, J. Thuburn, and C. J. Cotter. Computational modes and grid imprinting on five quasi-uniform spherical C grids. *Monthly Weather Review*, 140(8):2734–2755, 2012.

[WWGK17] N. Wintermeyer, A. R. Winters, G. J. Gassner, and D. A. Kopriva. An entropy stable nodal discontinuous Galerkin method for the two dimensional shallow water equations on unstructured curvilinear meshes with discontinuous bathymetry. *Journal of Computational Physics*, 340:200–242, 2017.

[ZRRB15] G. Zängl, D. Reinert, P. Rípodas, and M. Baldauf. The ICON (ICOsahedral Non-hydrostatic) modelling framework of DWD and MPI-M: Description of the non-hydrostatic dynamical core. *Quarterly Journal of the Royal Meteorological Society*, 141(687):563–579, 2015.